Tag Cloud Generation for Results of Multiple Keywords Queries

Martin Leginus, Peter Dolog, and Ricardo Gomez Lage

Department of Computer Science, Aalborg University,
Selma Lagerlofs Vej 300, 9220 Aalborg-East, Denmark
{mleginus,dolog,ricardo}@cs.aau.dk
http://iwis.cs.aau.dk/

Abstract. In this paper we study tag cloud generation for retrieved results of multiple keyword queries. It is motivated by many real world scenarios such as personalization tasks, surveillance systems and information retrieval tasks defined with multiple keywords. We adjust the state-of-the-art tag cloud generation techniques for multiple keywords query results. Consequently, we conduct the extensive evaluation on top of three distinct collaborative tagging systems. The graph-based methods perform significantly better for the Movielens and Bibsonomy datasets. Tag cloud generation based on maximal coverage is more suitable for the Delicious dataset because of the different statistical properties of the dataset.

1 Introduction

Tag cloud is an information retrieval interface that is commonly utilized by users of social tagging systems. It depicts a selection of terms used for resource annotations by the users. Tags of tag cloud are usually depicted with different colours and font-sizes. These visual aspects of tags express their popularity and importance within a system. A tag of the tag cloud links to a set of resources that are annotated and referenced with the given tag. Users can retrieve the set of resources by clicking on the relevant tag.

Tag clouds are usually studied as exploration interfaces depicting the most frequent tags of all resources in a system. In that sense, tag clouds provide a rough summary of resources and their topic distribution within the system. Therefore, the tag cloud is suitable for unspecific retrieval and exploration tasks and serves as a starting point for more specific keyword-based search [1].

In this paper we focus on tag cloud generation techniques for multiple keyword query results. In other words, as a user expresses his information goal with a combination of keywords a tag cloud is generated on top of all the related resources i.e., resources retrieved by the placed query. The tag cloud generation conditioned by multiple keyword query is motivated by several real-world scenarios.

Firstly, there is a need for query specific tag clouds due to the large amount of various resources within the recent collaborative tagging systems e.g., Flick

F. Daniel, P. Dolog, and Q. Li (Eds.): ICWE 2013, LNCS 7977, pp. 233–248, 2013.

(picture sharing site), Youtube (video repository) or Delicious (bookmark sharing system). The motivation for such tag clouds is:

- Underlying dataset contains a vast amount of data with diverse topics. Many of these topics are completely irrelevant for given users and the context of their work.
- Users would like to get an overview or follow dynamics of resources related to their interests or preferences (expressed in a query or in a user profile).

Secondly, the majority of information retrieval tasks are defined as a combination of several independent keywords. For instance, almost 67% of all placed queries in the USA are queries that consist of two or more keywords [3]. Similar statistics are reported for other countries. Therefore, there is a need for tag clouds generated with respect to multiple keywords queries.

In such scenarios, tag clouds present tags with respect to the query keywords entered by a user. This visual interface describes resources retrieved by the query that contain tags matching query keywords. Even though users can perform the same tasks as with most-frequent-tags tag clouds i.e., impression making, browsing or exploring resources, only a subset of resources related to the query term is considered for the tag cloud generation. Consequently, it results in an improved exploration of resources as it filters out irrelevant documents.

In this paper we propose a set of tag selection techniques to generate tag clouds from the results of multiple keyword queries. These techniques are extensions of existing methods that were introduced in [9] and [6]. The contribution with respect to our previous work [6] are as follows:

- Different encodings of prior probability distributions for stochastic restarts of random walk based algorithms. This was needed to achieve a better performance of graph based methods when multiple keyword queries are considered.
- A graph transformation is conducted on top of syntactically pre-clustered tag space. It results into higher coverage of the generated tag clouds as was shown in [7].
- The extended graph-based methods improve relevancy of tag clouds *with 45 % on Movielens* dataset and *21 % on Bibsonomy* in comparison to the state-of-the-art tag selection techniques.

We conclude through extensive experimental evaluations of three different datasets which method performs the best.

The structure of the paper is organized as follows; Section 2 positions our work with respect to related findings in the literature. Section 3 describes various tag selection techniques we have explored for the generation of tag clouds. Section 4 describes experimental set up and results which we gained from the experiments. Section 5 summarizes the paper achievements and roadmap to future work.

2 Related Work

Tag clouds are suitable for data-impression tasks within systems that contain annotated resources. The interface is generated from certain tag labels of system

resources. Users exploit tag clouds for the better understanding of a topic's diversity of a large number of considered resources. Tag clouds are often utilized within social networks and blog monitoring, medical-surveilance and fraud-detection systems.

Many studies investigated various aspects of tag cloud's visualization [2]. Tag color, size and position are important properties that influence user decision during tag cloud exploration. Furthermore, tags can be sorted alphabetically or sementically grouped based on their co-occurences. [7] proposes to cluster syntatically similar tags to avoid redudancies in the tag cloud which improves the coverage of generated tag clouds.

[9] proposes a synthetic user model aimed for tag cloud evaluation. They introduce various synthetic metrics that measure the qualities of tag clouds such as coverage, overlap and balance. Moreover, [9] introduces 4 tags selection algorithms for tag cloud generation. The first method selects the most popular tags within a system. The other two algorithms choose tags based on term frequency-inverse document frequency (tf-idf). The last and the most promising algorithm maximizes the coverage of selected tags in a tag cloud. These selection algorithms only focus on specific properties and aspects of tag cloud as coverage, frequency, and diversity. In this work, we consider extensions of these tags selection methods for multiple keywords queries.

As we presented in [6], coverage and overlap are too general measures and might not be suitable for query conditioned tag clouds. In this work, we exploit extensions of a topic sensitive version of PageRank algorithm, personalized HITS and constrained version of Markov Chain algorithm described in [6, 10]. The extension for tag cloud generation of multiple keyword query results is that the prior probabilities have to be set differently than in the case of one keyword queries to achieve better performance.

3 Methods for Tag Clouds Generation with Respect to Multiple Keyword Queries

In this section, we present a set of methods for tag cloud generation with respect to multiple keyword queries. For convenience, from now on we refer to multiple keyword query simply as query. The general application of these methods can be summarized as follows:

1. Retrieve all system resources annotated or related to the entered query.
2. Perform tags selection method on top of the tag space of retrieved system resources.
3. The top-k most relevant tags are presented in the generated tag cloud to the end user.

The following methods explore all tags used for annotation of resources that were retrieved with respect to the query. The exploration involves an evaluation of different aspects of invidual tags. The first group of presented methods select tags based on their coverage or populatity of resources in the system. The second

group of the methods transforms tags and their co-occurrence (a number of resources that are annotated with the same two tags) into a graph structure. The graph is then used as an input for different tag importance estimation algorithms.

3.1 Most Frequent Tags from Corpus (MFTC)

This method is the most common approach for generating tag clouds on top of the entire dataset. The entire tag space is sorted and orded according to the tag annotation frequency in descending order. The final tag cloud contains the top-k most frequent tags. The advantage is that the most frequent topics within the system are propagated to the tag cloud. On the other hand, the tag cloud does not cover other not so frequently represented topics which could be relevant for the user.

3.2 Most Frequent Tags from Query Result Set (POP)

The method is very similar to the previous one. The difference is that the set of resources is constrained to those resources that are annotated by at least one keyword from the query. The method creates tag clouds from top-k most popular tags of the documents (D_{T_q}) that are associated with the set of keywords of the query T_q. We assume that for each keyword exists the same tag in the system. The method was initially proposed by Venetis et al. [9] for single keyword queries. Drawbacks of this method are similar as with the MFTC technique.

3.3 Term Frequency - Inverse Document Frequency Selection (TFIDF)

The method ranks each tag t of the documents (D_{T_q}) that is associated with the query keywords T_q. The ranking function computes term frequency - inverse document frequency (tf-idf) for each tag and the document from the set of resources $D_t \cap D_{T_q}$ where D_t is the set of resources that are annotated by the tag t. These values are aggregated with the summation and are sorted in descending order. The top-k tags with the highest score are selected for the final tag cloud. The method was introduced by Venetis et al. [9] for single keyword queries tag cloud generation. The advantages and shortcommings of TFIDF based methods are similar to their advantages and disadvantages with respect to information retrieval tasks on top of traditional document repositories such as no consideration of semantic similarities between tags.

3.4 Max Coverage Selection (COV)

This selection explores a tag space of the documents (D_{T_q}) in the greedy fashion such that it tries to maximize a coverage of the selected tags. It iterates through the tag space and at each iteration step selects the tag that covers the highest number of uncovered documents. The method was proposed by Venetis et al.

[9] for single keyword queries. The advantage of the method is maximization of coverage and at the same time minimization of overlap between tag clouds tags. However, in our previous work [6], we pointed out that there are problems with tag cloud generation that maximizes coverage. The optimization of coverage might result into the generation of tag clouds that contain terms with high coverage but are irrelevant for the specific user's information retrieval goal.

3.5 Graph Based Methods

In the following subsections, we describe graph based tag cloud generation methods. Firstly, we introduce graph transformation of the tag space. Secondly, we describe three graph based methods. Finally, we propose a new adjustment of prior distribution for the random restarts of graph-based methods.

Graph Creation. The problem of a tag cloud generation with respect to specific query tags can be also transformed into estimating relative importance of other tags within underlying graph of tags with respect to the query. Firstly, an original tag space can be transformed into a graph structure where different aspects of tags relations can be captured.

In this work we utilize the following approach for graph creation. The graph creation is similar to our previous work [6], the difference is that the original tag space is syntactically pre-clustered. The syntactical clustering is presented in Section 3.10. First, we calculate a tag pair co-occurence using Jaccard similarity coefficient for all clustered tags, (see Formula 1) where $\mathrm{cocr}(t_i, t_j)$ represents co-occurence of two tags t_i, t_j i.e., the number of resources annotated by both tags t_i, t_j. Further, $\mathrm{f}(t_i)$ denotes a frequency of use of the particular tag in the system).

$$\mathrm{JAC}(t_i, t_j) = \frac{\mathrm{cocr}(t_i, t_j)}{\mathrm{f}(t_i) + \mathrm{f}(t_j) - \mathrm{cocr}(t_i, t_j)} \tag{1}$$

When the calculated similarity for a tag pair is greater than a predefined threshold α, we consider such tags as similar. Second, each similar tag pair is transformed into two directed edges $t_1 \to t_2$ and $t_2 \to t_1$. One could propose to construct only undirected graph, but in this context we construct directed graph in order to apply various graph based algorithms which are limited only to directed graph structures. Finally, we employ various graph-based algorithms to select the most relevant tags with respect to query tags.

In order to introduce different graph algorithms for relative importance of tags within the graph, we present preliminaries on graphs and their properties.

Graph preliminaries A directed graph $G = (V, E)$ consists of two sets: a set of nodes V and a set of edges E. In this context, each node corresponds to one particular tag from the underlying clustered tag space (the clustering is described in 3.10). Each edge e is defined as an ordered pair of nodes (u, v) for directed link from u to v. Therefore, an edge represents a relationship between two particular tags. A walk from u to v is a sequence of edges $(u, u_1), (u_1, u_2) \ldots (u_k, v)$. A walk is a path if no nodes are repeated. Various graph algorithms are based on

a notion of k-short paths. It is a set of all paths shorter than k between u and v in the graph. Other algorithms use a number of outgoing and ingoing edges for deriving an importance of a particular node in the graph. Therefore, we define

- $s_{out}(u)$ as a set of distinct outgoing edges from u
- $s_{in}(u)$ as a set of distinct ingoing edges towards u
- $d_{in}(u) = |s_{in}(u)|$ and $d_{out}(u) = |s_{out}(u)|$

Graph Based Methods. We present three graph-based algorithms for estimating a relative importance I of each node in the graph with respect to the set of query keywords. These methods are originally introduced in [6] for the tag cloud generation conditioned by single keyword queries. The algorithms rank an importance of a tag t with respect to the query keywords T_q where $\{t, t_q\} \in G$ and $t_q \in T_q$. It is denoted as:

$$I(t|T_q)$$

Sorting tags according to their importance with respect to the tags from T_q yields ranked tags. Eventually, it is easy to proceed with a tag cloud generation when only top-k most relevant tags are selected for the final tag cloud.

There exist two distinct approaches for estimating a relative importance of tags.

1. **Distance based approach:** The intuition is that a node is more relevant to a particular node when a graph distance between these nodes is smaller. In this context, a tag t is less relevant to the set of query tags T_q when there are many intermediate tags on the graph path from t to tags from T_q.
2. **Stochastic approach:** The importance of a node is estimated with a stochastic process. A reader can imagine a token that is randomly traversing a graph. The token steps from one node u to another node v with a transition probability which is given by $d_{in}(v)$ and $d_{out}(v)$. The random traversal of the graph after a certain time converges. It means that the time of the token spent at a certain graph node become stable. The time spent by the token at each vertex expresses an importance of the particular vertex in the graph.

In this paper, we focus on the stochastic approach only. This is motivated by the finding from our previous work [6]. The distance based techniques are computionally expensive and directly dependent on the graph size.

Stochastic Approaches for Tag Cloud Generation. The technique of measuring importance of nodes in the graph is based on the simulation of stochastic process e.g., random traversal of the graph. For the tag cloud generation it can be conceived in the following way: There is a tag cloud creator that randomly explores the graph structure of tags. The tag cloud creator visits a particular node (tag) in the graph and then randomly hops to one from the tag neighbours of the current node. Each visit of the cloud creator at certain tag can be interpreted as the evaluation whether a given tag is important for the desired

tag cloud. When this stochastic process lasts infinitely long time, a period that the tag cloud creator has spent at a certain tag can be perceived as its importance. As imaginary tag cloud creator can stay within a particular part of the sub-graph too long a definition of a back probability needs to be included. The back probability influences how often the random traversal of graph should be restarted i.e., start a new random walk from one node that belongs to the query keywords set. The transition probability from a tag t_1 to t_2 is defined as

$$p(t_2|t_1) = \frac{1}{d_{out}(t_1)}$$

for all tags t_2 that have an ingoing edge from t_1. Otherwise, a transition probability equals 0. This process can be classed as a first-order Markov Chain and there are various algorithms based on this process.

3.6 PageRank with Priors (PgRank)

This famous algorithm has been proposed in [8] and since then it has been used for relevance and importance ranking of web resources within the web graph. The PageRank reflects a behaviour of a random surfer where a certain set of similar web pages is browsed by the user. After some time, surfer randomly visits a different web page. Let us assume that the random surfer will browse and explore web pages forever, in such situation, a time that he spends at a particular site expresses also its importance in the web graph.

In this work, we focus on estimation of an importance of the tag t with respect to T_q. Therefore, we describe a topic sensitive PageRank as was introduced by [4]. A bias towards query tags is introduced with a vector of prior probabilities $p_r = \{p_1 \dots p_{|V|}\}$. Sum of prior probabilities equals to 1. In this work we consider more query tags T_q, hence the prior probabilities are set only for these query nodes. The setting of prior probabilities is described in the following Section 3.9. A random restart of the random walk is achieved with a back probability β. It conditions how frequently a stochastic process returns back to the root nodes i.e., query tags from T_q. Consequently, we are able to define iterative stationary probability:

$$\pi(v)^{(i+1)} = (1-\beta)\left(\sum_{u=1} d_{in}(v)p(v|u)\pi^{(i)}(u)\right) + \beta p_v \tag{2}$$

The resulting importance ranks biased towards T_q are considered as definition of importance after convergence i.e.;

$$I(t|T_q) = \pi(t) \tag{3}$$

The advantages of graph-based methods were presented in [6], where more relevant tag clouds were generated. However, the method requires to set up several parameters such as the back probability β and prior probabilities with respect to a specific dataset.

3.7 HITS with Priors (HITS)

In the similar way, we utilize HITS algorithm where a bias towards root query nodes is introduced through a vector p_r of prior probabilities.

Similarly, the setting of the prior distributions is described in Section 3.9. A random surfer is achieved with a back probability β - it determines how often we jump back to a root node. The iterative stationary distributions for authorities and hubs is defined in the following way:

$$a(v)^{(i+1)} = (1-\beta)\left(\sum_{u=1} d_{in}(v)\frac{h^{(t)}(u)}{H^{(i)}}\right) + \beta p_v \tag{4}$$

$$h(v)^{(i+1)} = (1-\beta)\left(\sum_{u=1} d_{out}(v)\frac{a^{(t)}(u)}{A^{(i)}}\right) + \beta p_v \tag{5}$$

where

$$H^i = \sum_{v=1}^{|V|}) \sum_{u=1} d_{in}(v)h^i(u) \tag{6}$$

$$A^i = \sum_{v=1}^{|V|}) \sum_{u=1} d_{out}(v)a^i(u) \tag{7}$$

The motivation is similar as with the topic sensitive PageRank algorithm only difference is that at each even step of the traversal only ingoing edges to the current node (tag) are considered and at each odd step only outgoing edges are considered as possible steps. The resulting importance ranks (stationary distribution of each node) biased towards T_q are considered as definition of importance after convergence i.e.;

$$I(v|T_q) = \pi(v) \tag{8}$$

The algorithm has the similar properties as the PageRank with priors or the k-step Markov Chain methods. However, the setting of the prior distribution is more complicated. There is a need to encode prior probabilities for hubs and authorities in the graph.

3.8 k-step Markov Chain (k-MarkovCh)

The technique is different to the previous methods in the implementation of a random surfer model. It is achieved with a path length limitation. The step constraint determines how often a token jumps back to one query node from the query keyword set. The smaller the path length parameter is the more often a stochastic traversal of the graph restarts back to a node chosen according to the prior probabilities of nodes. We discuss this issue in the following Section 3.9. A bias to the root nodes is introduced through a vector p_r of prior probabilities.

The constructed graph can be represented as well with the transition matrix A which is constructed in the following way. If there is an directed edge from a node u to v, then we put a transition probability on row u, column v of the matrix A.

$$I(v|R) = [A \cdot p_R + A^2 \cdot p_R \ldots A^K \cdot p_R] \tag{9}$$

The resulting importance ranks (stationary distribution of each node) biased towards T_q are considered as definition of importance after convergence i.e.; The method depends on the proper setting of the number of steps the random walk is performed until the restart is performed. The tuning of this parameter might be complicated and vary for different datasets.

3.9 Adjustment of Prior Distribution for Stochastic Restarts of Random Walks

The prior probability for single keyword queries is defined in a simple, straightforward way. The probability for the query tag t_q is set to 1 and for all other nodes it is set to 0 [6]. However, this approach does not work for multiple keyword queries. The naive extension would be to set a prior probability for each query tag t_q from the set of query tags T_q such that $p(t_q) = \frac{1}{|T_q|}$ However, this approach does not capture individual popularity of a query tag in the corpus. In other words, when a rarely used tag is chosen as a query tag t_q, such tag does not co-occure with many tags. Therefore, there are not many edges connecting this graph node with other nodes. Therefore, a random traversal of the graph initiated from the rarely used tag/node might reach not important/relevant nodes (tags). Consequently, it results into an inclusion of irrelevant tags into the tag cloud. We verified this assumption by series of preliminary evaluations. The naive approach of setting prior probabilities for multiple keywords queries suffers from the inclusion of irrelevant tags into the final tag cloud. Therefore, we

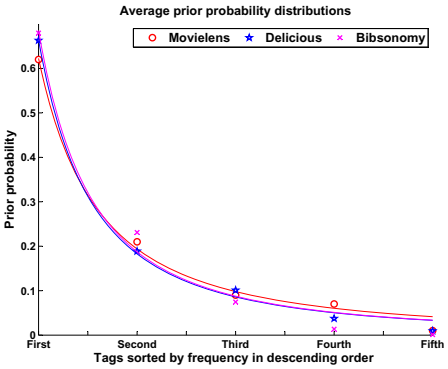

Fig. 1. Prior probability distributions and their corresponding exponential fit for Movielens, Delicious and Bibsonomy datasets when generating tag clouds for queries that consist of five keywords

propose a simple intuitive setting of prior probabilities which capture a relative popularity of the individual query tag t_q from the set of query tags T_q. The relative popularity for each query tag t_q is computed in the following way:

$$p(t_q) = \text{popularity}(t_q) = \frac{f(t_q)}{\sum_{t_{qi} \in T_q} f(t_{qi})}$$

The sum of all popularities of tags from T_q equals to 1. Therefore, we set for each query tag t_q, the prior probability that equals to popularity(t_q). In this way, the more popular tags are more likely to be chosen after the restart of the random walk. To motivate the presented approach, we observed the frequency distribution of query tags, when the size of the query tags is set to 5. We randomly selected 30 different queries for each dataset, each query consists of a set of five query tags that are semantically similar (the selection process is more detailed in Section 4.2). Figure 1 presents an average distribution of relative popularities of query tags for each dataset. All three presented distributions and their corresponding exponential fits indicate a large differences in relative tags popularities among considered query tags. Therefore, it empirically proves our intuition about adjusting prior probabilities of query tags with respect to their relative popularity.

3.10 Syntactical Pre-clustering of Tags

All the presented techniques generate tag clouds from the syntactically clustered tag space. We compute Levenhstein edit distance for each tag pair from the initial tag space as it was succesfully utilized in our previous work [7]. Once, an edit distance is calculated, the tag space is split into clusters. Each cluster consists of tags where the Levenhstein distance is equal or lower than a defined threshold (a number of maximum changes to transform a tag from the tag pair into a second tag). Then, the most frequent tag for each cluster is selected and is used in all further computations. It represents all other tags from a considered cluster. This syntactical pre-clustering avoids redudancies in the generated tag clouds, results into a denser graph structure for graph based methods.

4 Experiments

To compare the presented tag cloud generation methods described in Section 3 we conduct the following experiments. We measure the relevance of generated tag clouds with respect to the queries. The queries consist of several (ranging from two till five keywords) context-related tags that were derived from user profiles. The motivation is to simulate different information retrieval scenarios such as retrieval goal defined by multiple tags, retrieval of resources similar to the selected resource which can be described by the set of assigned tags, and various surveilance tasks where an expert monitors occurrence of predefined terms in the system. The relevance of the generated clouds is measured on top of the

Movielens dataset,a snapshot of Bibsonomy dataset [5], and Delicious dataset. The Bibsonomy dataset contains 206589 distinct items and 51565 tags. The total number of tagging posts is 466818. The Movielens dataset contains 16518 unique tags and 7601 movies. The total number of tagging posts is 95580. The Delicious dataset represents all bookmarking activities on www.delicious.com from 8th till 16th of September 2009. It contains 187359 users, 185401 unique tags and 355525 bookmarks. The total number of tagging posts is 2046868.

The tag cloud selection techniques are implemented in Java 6 and source code together with all results are available on our Web site[1].

The rest of the evaluation section is organized as follows. Firstly, we define required evaluation metrics. Secondly, we describe a methodology of the conducted experiments. In the end, we present and analyze evaluation results.

4.1 Evaluation Metrics

In this paper we measure the *relevance* of generated tag clouds with respect to the queries that consist of several context-related tags. We do not measure *coverage*, as we showed in our previous work [6] as this synthetic metric might be misleading. The following paragraph introduce formally the *relevance* metric.

A set of exiting documents is denoted as D , the whole set of existing tags is denoted as T, and the set of documents assigned to a tag $t \in T$ is denoted as D_t. The generated tag clouds is a set of tags which is denoted as T_c. The relevance of T_c expresses how relevant the tags in T_c are with respect to the query keyword t_q. We compute a relevance of each tag t from T_c in the following fashion:

$$rel(T_c) = avg_{t \in T_c} \frac{|D_t \cap D_{T_q}|}{|D_t|}, \tag{10}$$

where $|D_t|$ is the number of documents assigned to a tag t and $|D_{t_q}|$ is the number of all documents that are associated with a query tag t_q. The metric ranges between 0 and 1. When the relevance for a particular tag t is close to 1, the majority of documents annotated with a tag t is covered by the documents from D_{t_q}. The more D_t and D_{t_q} overlap, the more related t is to t_q. When $D_t \subseteq D_{t_q}$, then t can be perceived as more specific sub-category of the original query t_q.

4.2 Evaluation Methodology

We randomly select 15 distinct users from each dataset. Each user profile is pruned such that it contains only context-related tags. In this evaluation, we define context-related tags as all the tags assigned by the user to the semantically similar resources. The intention is to avoid tag cloud generation for semantically different keywords in the query. For example, a tag cloud generated with respect to the query keywords like *C*hristianity and Ubuntu is likely to not produce any relevant results to the users. Therefore, we concentrate only on tags which are topically similar e.g., *r*ussian, dictionary and software. We assume that all

[1] `http://people.cs.aau.dk/$\sim$mleginus/icwe2013/`

resources annotated by the given user are semantically similar when they share at least one tag. For each user, we iteratively change the size of the pruned user profile and measure the relevance of the generated tag clouds.

For each query set of tags T_q, where the size of T_q equals k, we perform the following evaluation:

1. Generate a tag cloud with respect to given query tags T_q utilizing specific tags selection method such that tag cloud contains at most n-tags.
2. Measure the *relevance* of the generated tag cloud.
3. Increase the size of tag cloud n. If maximum size is reached, increment the size of T_q.

The above-described evaluation is conducted for each datasets and all considered tags selection methods.

4.3 Graph-Based Techniques

Graph-based tag clouds generation consists of these two steps:

1. Perform syntactical clustering of the original tag space.
2. Make a graph transformation from the syntactically clustered tag space.
3. Make a tag selection utilizing graph-based relevancy ranking algorithms with respect to the set of query tags T_q.

Each phase requires a certain parameters setting which are presented in the following paragraphs.

Graph Creation from Tag Space. An important step of graph-based tag cloud generation is a proper graph transformation. The proposed approach is computing a tag pair co-occurence for all tags. When this measure for a tag pair is greater than a predefined threshold α, we consider such tags as similar. Eventually, each related tag pair is transformed into two directed edges $t_1 \rightarrow t_2$ and $t_2 \rightarrow t_1$. We have manually explored various similarity thresholds for both datasets and attained the best results with $\alpha = 0.035$ for Bibsonomy, $\alpha = 0.2$ for Movielens and $\alpha = 0.01$ for Delicious.

Parameters Setting of Graph-Based Techniques. The performance of *Pagerank (PgRank)*, *Hits* and *k-step Markov Chain (k-MarkovCh)* strongly depends on proper parameter settings. As the goal of this work is to identify the most relevant tags with respect to a given query tag t_q, we set the parameters in the following way:

- Prior probabilities for all algorithms are defined as relative ratio of frequency t_q with respect to the total frequency sum of all the tags of the query, and 0 for other tags.
- A back probability β for *Pagerank* and *Hits* is relatively high ($\beta = 0.9$) in order to introduce often restart of random walk from the query tag.
- For *k-step Markov Chain* method, we set a constant k to 6.

The majority of these parameters were the same as in [6, 10].

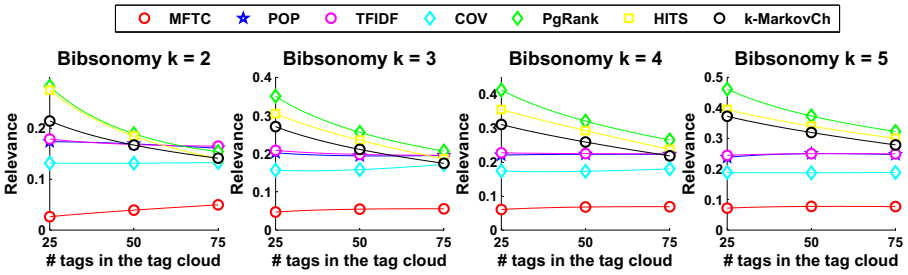

Fig. 2. Relevance on Bibsonomy dataset with different selection techniques and their corresponding logarithmic fit

4.4 Results

We conducted the evaluation for the presented datasets and the methods following the introduced methodology. We iteratively increased the number of tags in the tag cloud starting with 25 till 75 tags with the step 25. Moreover, we explored different query sizes, i.e., number of tags in the query, ranging from $k = 2$ till $k = 5$. The results for the Bibsonomy dataset are presented in Figure 2. The graph-based tags selection methods outperform all baseline techniques in almost all settings. The largest improvements can be observed for tag clouds with the size $n = 25$ and all query sizes. In these cases, the Relevance is improved about 0.21, 0.19, 0.15 and 0.18 for the query sizes $2, 3, 4$ and 5 respectively. The MFTC method attains the worst results, similarly COV method perform worse than other compared methods. Obviously, tag cloud optimization with respect to the coverage results into inclusion of more irrelevant tags to the final tag cloud. The best graph based method is Pagerank algorithm. The performance of HITS algorithm decreases as the number of query tags increases. Similarly, k-Markov Chain attains lower Relevance as PageRank algorithm, because k constant is still the same for larger query sets. Therefore, the imaginary token may reach further from the query tags in the graph. However, this can be beneficial for cases where the goal is to deliver more diverse tag clouds as overlap is slightly lower than for Pagerank algorithm. Graph based methods attain the best results also for the Movielens dataset. The Relevance is improved about 0.40, 0.44, 0.45 and 0.3 for these query sizes $2, 3, 4$ and 5 respectively (see Figure. 3). The TF-IDF methods does not attain such good results as there are many relevant tags which are used frequently on top of all resources in the dataset. The graph based methods perform very similarly. However, the methods attain lower relevance for large tag clouds.

On the contrary, the graph based methods attain similar or lower relevance for tag clouds generated from the Delicious dataset (see Figure 4). For tag clouds which contain 25 tags the best performing method is PageRank algorithm. On the other hand, for the tag clouds with more tags, the COV method outperforms other techniques. Decreased performance of the graph based methods is caused by the Delicious dataset data distribution. In particular, there are many very

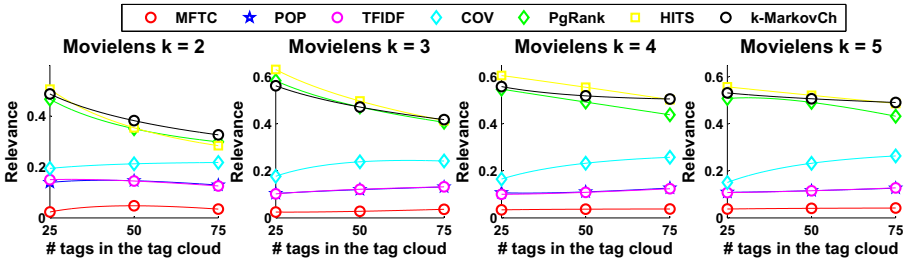

Fig. 3. Relevance on Movielens dataset with different selection techniques and their corresponding logarithmic fit

frequent tags in the dataset, i.e., almost 20 tags that were assigned at least 10000 times, almost 500 tags that were placed by users at least 1000 times. On the other hand, there are 172554 tags that were utilized for annotation less than 10 times. Consequently, the underlying co-occurrence graph links very frequent tags with very rarely used tags. It results into the inclusion of more frequent tags into tag clouds. Such inclusion causes lower relevance.

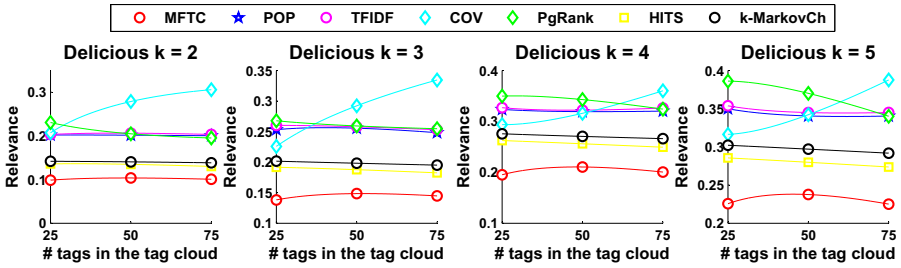

Fig. 4. Relevance on Delicious dataset with different selection techniques and their corresponding logarithmic fit

4.5 Discussions and Limitations

The presented results demonstrate that graph-based methods can be succesfully exploited for tag cloud generation tasks. The improvements are significant (in some cases almost tripled relevance such as for the Bibsonomy and Movielens dataset). The advantage of the graph-based methods is an ability to generate tag clouds with respect to the particular set of query tags. The methods allow you to predefine which tag from the query tags should be more preferred by adjusting prior distribution for stochastic restarts of Markov Chains. Moreover, these methods allow you to generate more diverse tag clouds with still relatively high relevance (the smaller back probability β is the more diverse final tag clouds are). Despite of these advantages, the methods do not perform that well on top of datasets with the long-tail distribution of tags. We consider datasets

with the long-tail distribution of tags as those that contain very few tags which are frequently utilized within the system. Moreover, these datasets include a large number of infrequently used tags. The graph-based methods select tags utilizing co-occurrence graph which links more frequent tags with rarely used tags. Obviously, it results in the inclusion of more frequent tags into the final tag clouds. This bevahiour causes decreased relevance of the generated tag clouds.

5 Conclusions

We explored the set of tag cloud generation methods with respect to multiple keyword query. The graph-based methods perform the best at the Movielens and the Bibsonomy datasets. This is achieved due to the proposed extension of the setting of prior probabilities for the random walk based algorithms. On the other hand, the graph-based methods do not perform well for the Delicious dataset. The Delicious dataset has different distribution of tags with many very frequent tags which decreases performance for more specific queries. For future work, we plan to investigate how to build tag clouds based on user preferences. Moreover, we aim to define tag cloud generation methods that will capture diversity and novelty of the considered resources.

Acknowledgements. This work has been supported by FP7 ICT project M-Eco: Medical Ecosystem Personalized Event-Based Surveillance under grant No. 247829. Moreover, the author wish to thank Vicki Chapman for her help with the proofreading.

References

1. Aras, H., Siegel, S., Malaka, R.: Semantic cloud: an enhanced browsing interface for exploring resources in folksonomy systems. In: Workshop on Visual Interfaces to the Social and Semantic Web (VISSW2010), IUI 2010, Hong Kong, China, 2010 (February 7, 2009)
2. Bateman, S., Gutwin, C., Nacenta, M.: Seeing things in the clouds: the effect of visual features on tag cloud selections. In: Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia, pp. 193–202. ACM (2008)
3. A.T. Company, Keyword and search engines statistics (2013)
4. Haveliwala, T.: Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. IEEE Transactions on Knowledge and Data Engineering 15(4), 784–796 (2003)
5. Knowledge and U. o. K. Data Engineering Group. Benchmark folksonomy data from bibsonomy, version of January 1, 2010 (2010)
6. Leginus, M., Dolog, P., Lage, R.: Graph based techniques for tag cloud generation. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media. ACM (2013)
7. Leginus, M., Dolog, P., Lage, R., Durao, F.: Methodologies for improved tag cloud generation with clustering. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 61–75. Springer, Heidelberg (2012)

8. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, Previous number = SIDL-WP-1999-0120 (November 1999)
9. Venetis, P., Koutrika, G., Garcia-Molina, H.: On the selection of tags for tag clouds. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 835–844. ACM, New York (2011)
10. White, S., Smyth, P.: Algorithms for estimating relative importance in networks. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 266–275. ACM, New York (2003)