Indirect Keyword Recommendation

André Sabino, Armanda Rodrigues, Miguel Goulão and João Gouveia
CITI, Departamento de Informática,
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa,
Quinta da Torre, 2829–516 Caparica, Portugal
{amgs@campus.,a.rodrigues@,mgoul@,j.gouveia@campus.}fct.unl.pt

***Abstract*—Helping users to find useful contacts or potentially interesting subjects is a challenge for social and productive networks. The evidence of the content produced by users must be considered in this task, which may be simplified by the use of the meta-data associated with the content, i.e., the categorization supported by the network – descriptive keywords, or *tags*. In this paper we present a model that enables keyword discovery methods through the interpretation of the network as a graph, solely relying on keywords that categorize or describe productive items. The model and keyword discovery methods presented in this paper avoid content analysis, and move towards a generic approach to the identification of relevant interests and, eventually, contacts. The evaluation of the model and methods is executed by two experiments that perform frequency and classification analyses over the Flickr network. The results show that we can efficiently recommend keywords to users.**

***Keywords*—Tagging, Social Networks, Social Graph**

I. Introduction

Helping users to find potential contacts or interesting subjects is a challenge for applications that support productive networks. Many systems provide awareness channels for suggestions to enhance the description of an item, and in order to suggest an interesting subject, applications analyze the content of the user's production. This content includes the data in each user's item, and the meta-data associated with it, i.e., the items' categorization and content description enabled by keywords. This work focuses on keyword analysis, rather than content analysis, avoiding the problems of processing heterogeneous content and access difficulties due to privacy and data sensitivity issues.

We define productive network as any network through which users share content, and are able to annotate that content. This definition includes social networks and networks that support cooperative work (CSCW). A key aspect of productive network is the presence of an annotation mechanism.

Annotations are a fundamental part of systems that deal with either user generated content, or which aggregate information according to preferences. They bridge the disciplines of personal information management, information architecture, and social software [1].

There is extensive work on keyword recommendation to enhance an item's description, usually focused on the keywords that are similar to those already used to describe the item. In [2], the authors address the problem with a frequency analysis of keyword co-occurrence. We explore a similar approach, with the goal of constructing a ranked list of keyword recommendations to the user, instead of a particular item. We look for keywords that are related with the keywords of the user's items, but which are not used, by the user, to categorize items. We define our approach through an information model that represents the social graph of the network. Ultimately, our list of keyword recommendations can be used to construct a list of user recommendations.

In this context, the contributions of this paper are:

- An information model that relates users, items and keywords;
- Two keyword recommendation strategies that use our information model. The first performs frequency analysis over the model, while the second one relies on the model features to train a classifier;
- A comparison of keyword suggestion ranking strategies.

The remaining of this paper is organized as follows. Section III presents the problem of finding interesting keywords to recommend. Section IV describes the model and the potential relationship extraction methods. Section V presents the case study used to evaluate our methods. Section VI presents a discussion of the results, and the outline of future experiments. Section II discusses the related work. Section VII draws some conclusions.

II. Related Work

Item annotations are very much related with the term folksonomy (folk, or user, generated taxonomy). Thomas Vander Wall coined this term in 2004 [1], and is used to describe the user-generated taxonomies that became a distinctive aspect of Web 2.0. To address the problem of lack of a clear and general semantic in folksonomy, Damme, et al. [3], propose the study of *folksontology*. The authors resume most issues we also identified with annotations in Flickr, and propose approaches to derive ontologies from folksonomies, promoting structure to enrich keyword semantics, and to integrate folksonomies and the semantic web. Xu, et al. [4], also discuss the use of keywords for the semantic web.

Contact suggestions and keyword recommendation are active research subjects. Roth, et al. [5], discuss a method to suggest contacts based on email contact lists and frequently used email addresses. Hecker, et al. [6], discuss how keywords are used and the motivation their adoption. Nov, et al. [7], focus on the motivation for annotating by Flickr users, and their keyword usage patterns.

[1] http://vanderwal.net/folksonomy.html

In [2], [8], [9], several authors present approaches for the design of keyword recommendation systems. The authors focus on strategies to recommend keywords to enhance items' descriptions, and usually those keywords already annotate items of the user's contacts. Lappas, et al. [10], discuss how social endorsement techniques can be used for keyword recommendation and ranking. Stefanidis, et al. [11], discuss the use of preference contexts for group recommendation systems.

Liu, et al. [12], discuss keyword ranking using a probabilistic approach, also using Flickr as a case study. Wang, et al. [13], present a machine learning approach for keyword ranking. These authors do not focus on keyword discovery.

Zhou, et al. [14], discuss several methods to recommend users in social annotation systems (social *tagging*). The case study used is the de.licio.us, and the approach is based on the proximity network of users. We directly compare our results with this approach.

Chi, et al. [15], also analyze the social annotation service del.icio.us to show that, in that system, annotating is primarily a method of personal organization, with individuals using a personal vocabulary while annotating. The authors state that, although there is a clear lack of structure in that and in similar social annotation services, hints of a global language emerge at some point in those networks.

Zubiaga, et al. [16], use information retrieval benchmarks to show that users whose keywords classify items outperform users whose keywords describe the content, which is compatible with our results, where a large number of keywords are associated with only one item, not serving as a classification system, but only as a descriptive one.

Leskovec, et al. [17], studied how social graphs evolve over time and identified the following characteristics, in the graph: the diameter of the graph tends to shrink over time; the number of its edges is a Power Law of the number of nodes, over time; it follows a pattern in which highly linked nodes have an improved chance of being reached by new nodes; the formation of local communities of nodes; and the tendency to contain a giant connected component. From these assertions, the authors proposed a model to generate graphs with such characteristics - the Forest Fire model, proved to be particularly suitable to sample social graphs [18]

III. Finding Interesting Keywords

This work addresses the recommendation of keywords for a user in a productive network, suggesting keywords related with the user's production. The goal is to promote interest discovery, and not to enrich a particular item's description. Ultimately, this list of keyword recommendations can be used to build a list of contact recommendations.

We present an example to provide context to the following sections.

A. Motivating Example

Our example concerns the task of suggesting the picture gallery of a Flickr [2] user to another user. Specifically, we are interested in suggesting users that do not share the same keywords.

In our example, we consider three Flickr users, Alice, Bob and Carol. Alice and Bob annotated some of their pictures with the keyword ”London”. In some of those pictures, Bob also used the keyword ”Knightsbridge”. Carol also used the keyword ”Knightsbridge” to annotate some of her pictures, but did not use the keyword ”London”. One can build a path of keywords to connect pictures and, ultimately, users. In our example, the shortest path between Alice and Carol is through keywords ”London” and ”Knightsbridge”. Our suggestion task is to present Alice with a list of keywords in the same conditions as ”Knightsbridge”, which enables the construction of a list of people in the same conditions as Carol.

Finally, we expect that several keywords meet the same conditions as ”Knightsbridge”, so the solution must rank the list of suggestions according to some criteria.

B. Problem Statement

The problem of recommending a keyword is defined as follows: For a particular user, based on a set of user defined keywords for her items, there is a set of candidate keywords for recommendation, which are not used by the user, and which can be ranked according to some strategy. In our example, the user we want to recommend keywords to is Alice, and the task would be to find all the keywords in the same conditions as ”Knightsbridge”.

There are two distinct tasks:

1) The first task is the definition of the set of keywords that are candidate for recommendation;
2) The second task is the construction of the ranked list from the candidate keyword set.

To address both tasks, we represent the information in the productive network, such as the Flickr network of our example, through a model that relates users, keywords and items. The model constructs graphs similar to social graphs [17]. Using keywords as nodes, the implicit graph of keywords relates two keywords if both are assigned to the same item.

From a user's perspective, we propose to use the implicit graph of keywords to build a set of candidate keywords that do not annotate any of the user's items, but are related with the keywords that do. The keyword recommendations expected from this strategy are particularly interesting to expose related keywords to the user, which can be explored by the application to offer suggestions for related subjects. In a scenario like our example's, keyword recommendations such as ”Knightsbridge” would enhance content discovery and user discovery.

We also use features extracted from the model to train a classifier for each user. The goal is to produce a set of candidate keywords, but focusing on suggesting new interests to the user.

Task 2 deals with the presentation of the information. In order to be useful, the candidate keywords need to be ranked. Finally, depending on the system, the top n elements of the list of ranked candidates is shown to the user. In our example, the Flickr interface would deliver a list of top candidates to Alice, which would be constrained by usability guidelines.

[2] https://www.flickr.com

From the ranked list of keyword recommendations is also possible to define a list of recommended users, selecting the user that most frequently used the keyword, filtering out duplicates, for each keyword in the list.

IV. Information Model and Keyword Discovery Methods

This section presents a model that represents information of social networks, and computer supported cooperative work networks – i.e., productive communities. The information is represented by three concepts: user, item and keyword. In this context we consider the following:

- A user is related with one or more items submitted and/or available in the network;
- A keyword is an expression (of one or more words) which may be used to categorize or describe the content of one or more items;
- An item is the result of an effort led by one or more users in the network, and whose content may be categorized or described by keywords.

Between these concepts, relationships are assumed from evidence taken from the data available in the network. A user is directly related to all of her items and, as items are associated to keywords, users will be automatically associated with the keywords they have used to categorize and describe their items. The model allows for the representation of three types of binary relationships between users:

- Co-authoring: when two users share the same item. This is not present on our example, but is possible on some productive networks;
- Direct relationship: when two users are not co-authors but their items are related with a common set of keywords (i.e., users share keywords, like Alice and Bob share ”London”);
- Indirect relationship: when there is no direct relationship between two users but some of the keywords used to categorize their items are related by other users' items, like the keywords ”London” and ”Knightsbridge” are related by Bob's pictures, therefore creating an indirect relationship between Alice and Carol.

To support the concept behind identifying indirect relationships, we propose that the frequent association of two keywords with common items has value for users of one of those keywords, e.g., for Alice, there is value in the association between ”London” and ”Knightsbrigde” in Bob's pictures.

The intuition behind our approach is that one keyword is potentially relevant to one user if it is frequently associated with items that other keywords of the user are associated with (excluding the items of the user). The value of this potential grows with the frequency of the association of the keywords, and can be ranked. This is what happens in our example: there are, potentially, many keywords in the same situation as ”Knightsbridge”, which could be used to suggest subjects and people to Alice. Our goal is to define a model and methods suitable to find and rank these keywords associations. The application of this process to several keywords in the same conditions results in a sorted list of keywords that are potentially interesting to the user and, ultimately, a sorted list of indirect relationships of the user.

The remaining of this section describes the details of the model, which include the formal representation of the concepts and relationships, the implicit social graph and, finally, the experiments designed to test the methods.

A. Formalization

This section gives a more precise definition of the model concepts that were introduced in the previous sections.

The basic elements of the model are users, U, items, I and keywords, K. Items are owned by users, and annotated with keywords.

Let us define U, I and K such as:

$$U = \{U_1, \cdots, U_n\} \textit{ is a finite set of users, } n \geq 1$$
$$I = \{I_1, \cdots, I_m\} \textit{ is a finite set of items, } m \geq 1$$
$$K = \{K_1, \cdots, K_l\} \textit{ is a finite set of keywords, } v \geq 1$$

Definitions 1 and 2 represent the basic item management operations that the network provides to its users.

Definition 1: The ownership by an user, U_i, of an item, I_o, is defined by:

$$O(U_i) = \{I_t \mid I_t \textit{ is owned by } U_i, I_t \in I, U_i \in U\}$$
$$Own(U_i, I_t) = \{I_t \in O(U_i)\}$$

Note that, for the sake of simplicity, in our model, an item can only be owned by one user. However, this representation does support co-authoring scenarios. We will return to this later in this section.

Definition 2: The annotation of an item, I_t, by a keyword, K_p, is defined by:

$$T(I_t) = \{K_p \mid K_p \textit{ is associated with } I_t, K_p \in K, I_t \in I\}$$
$$Annotate(I_t, K_p) = \{K_p \in T(I_t)\}$$

We refer to keywords that are used in annotations as the user's *direct keywords*. In definition 2, the set $T(I_t)$ is the set of direct keywords of item I_t.

Definition 3: For a user, U_i, the set of all direct keywords of all of the user's items is defined by:

$$UK(U_i) = \{K_p \mid \forall I_t \in O(U_i), \forall K_p \in T(I_t)\}$$

We now define the relationships that items and keywords enable between users. We begin with the definition of direct relationship, which establishes a link between users.

Definition 4: A direct relationship, DR, between two users, U_i and U_j, is defined by:

$$\begin{aligned} DR(U_i, U_j) = \{K_p \mid I_t \in I, I_u \in I, \exists K_p \in K :& \\ Own(U_i, I_t), Own(U_j, I_u),& \\ Annotate(I_t, K_p), Annotate(I_u, K_p)\}& \end{aligned}$$

It is now possible to describe the graph that is implicitly defined by the network.

Definition 5: The graph that relates users in the network is defined by $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, such that:

$$\begin{aligned} \mathcal{V} &= \{U_i \mid \exists U_j \in U : \mathit{DR}(U_i, U_j) \neq \{\emptyset\}\} \\ \mathcal{E} &= \{K_p \mid \exists U_i, U_j \in U : K_p \in \mathit{DR}(U_i, U_j)\} \end{aligned}$$

Note that the definition excludes isolated users, which cannot be related with any other user through any keyword. Our suggestion mechanism does not target users in isolation.

Based on definition 4, we now define indirect relationships.

Definition 6: A indirect relationship, IR, between two users, U_i and U_j, is defined by:

$$\begin{array}{l} \textbf{if} \\ \qquad \mathit{DR}(U_i, U_j) = \{\emptyset\}, \\ \qquad \exists U_k \in U, \exists K_p, K_q \in K : \\ \qquad\qquad K_p \in \mathit{DR}(U_k, U_i), K_q \in \mathit{DR}(U_k, U_j) \\ \textbf{then} \\ \qquad \mathit{IR}(U_i, U_j) = \{K_q \mid K_q \in \mathit{DR}(U_k, U_j)\} \\ \qquad \mathit{IR}(U_j, U_i) = \{K_p \mid K_p \in \mathit{DR}(U_k, U_i)\} \end{array}$$

In the graph of definition 5, indirect relationships refer to shortest paths of size two. These shortest paths are always defined by one keyword used by the user – a direct keyword –, and one that is not – an indirect keyword. Corollary 1 describes the set of indirect keywords, which are used to define indirect relationships.

Corollary 1: Definition 6 enables the definition of a set of indirect keywords, IK, of a user, U_i, such that:

$$\mathit{IK}(U_i) = \{K_p \mid K_p \in \mathit{IR}(U_i, U_j), \forall U_j \in U, U_j \neq U_i\}$$

For each user, U_i, the goal is to find users, U_j, such that $\mathit{IR}(U_i, U_j) \neq \{\emptyset\}$. Our method is to build the set of indirect keywords of user U_i, which enables the construction of a list of users that use those keywords but do not have a direct relationship with U_i, i.e., the indirect relationships.

We calculate the number of items associated with each keyword in the list of indirect keywords. This value will be used to sort the list.

Definition 7: For a user, U_i, the list of indirect keywords with rank values, IK_r, is defined by:

$$\begin{aligned} R(K_p) &= \{I_t \mid \mathit{Annotate}(K_p, I_t)\} \\ \mathit{IK}_r &= \{\langle K_p, |R(K_p)| \rangle \mid K_p \in \mathit{IK}(U_i)\} \end{aligned}$$

The sorted list of indirect keywords is a reversed total order of IK_r.

The rank value for each keyword, as described in definition 7, is one approach among many. For instance, the total number of keywords that share an item with the indirect keyword could be used to calculate a rank value. We found that the number of items annotated with the keyword was best suited, and we show the results of an experiment which compares these methods in section VI-A.

Ultimately, after the list of indirect keywords is found, the goal would be to deliver a list of indirect users. In this work we focus on the validation of the ranked indirect keywords list, and we discuss the results of a set of experiments that are designed to evaluate the suitability of this list. Furthermore, we designed a learning method, evaluated by the experiment in section IV-B2, which is set to discover indirect keywords through a trained classifier. Section VI-B shows the results of that experiment.

Definition 1 states that, for the sake of simplicity, each item can only be owned by one user. Figure 1 presents a partial graph with our approach to co-authoring. Users U_2 and U_4 have a direct relationship. Users U_1 and U_4 have an indirect relationship, on which we focus our discussion. Given that our goal is to find indirect relationships, we cannot exclude the direct relationship between U_1 and U_2, defined by $I_{1_{U_2}}$. U_4 is a candidate for indirect relationship with U_1 through U_2 and U_3. If we exclude the path through U_3 and do not consider the direct relationship between U_1 and U_2, there are not enough conditions to define the indirect relationship between U_1 and U_4.

If we took a different approach, and considered that co-authors share direct relationships, then U_1 and U_4 would have a direct relationship, which cannot be assumed for an arbitrary network.

Our approach represents co-authored items by creating a copy of the item for each co-author. Copying items and defining direct relationships between co-authors provides assurance that the approach does not overlook any valid candidate for indirect relationships.

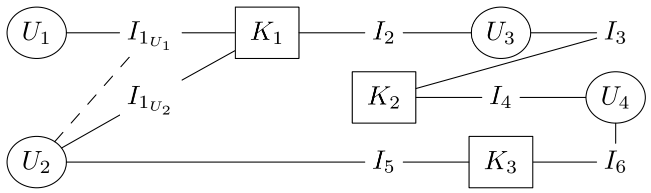

Fig. 1. Partial graph illustrating the representation of a shared item. It represents users in squares, keywords in circles and items without decoration. Although $I_{1_{U_1}}$ and $I_{1_{U_2}}$ both represent the same item in the network, they are actually distinct in the graph. $I_{1_{U_2}}$ defines the missing edge between U_2 and $I_{1_{U_1}}$ – represented by a dashed line.

B. Experiments

This section presents two experiments, both designed to evaluate the construction of the indirect keywords' ranked list. We propose that if a user annotated items with a keyword, that keyword would be a valid suggestion in a scenario where the network was modified so that that keyword becomes indirect to the user. Therefore, the strategy of both experiments is to remove one keywords from the set of keywords associated to the user. This method creates a new set of keywords that annotate the user's items, and the experiment outputs the ranked list of indirect keywords. Figure 2 presents the outline of the experiments for a particular user, U_i.

Given that the user explicitly annotated items with the removed keyword, we are sure that it is relevant to the user. Therefore, as presented in figure 2, the experiments' goal is to twofold: first, recover it as an indirect keyword; second, attribute a high rank value. Success in this setup implies that the model is capable of discovering interesting keywords.

```
Require: $O(U_i) \neq \emptyset$
for all $I_t \in O(U_i)$ do
    if $T(I_t) \neq \emptyset$ then
        for all $K_p \in T(I_t)$ do
            $T'(I_t) = \{K_q \mid K_q \in T(I_t), K_q \neq K_p\}$
            $I\!K_r = experiment(T'(I_t))$
            print $K_p \in I\!K_r$?
            print $rank(K_p, I\!K_r)$
        end for
    end if
end for
```

Fig. 2. Experiment outline. It shows the removal of the association between the user and keywords, that the experiment is designed to recover. The $rank$ function returns the position of K_p in $I\!K_r$.

1) Frequency Analysis: Our first method is to use a frequency analysis to discover keywords that are indirectly related with the user. To set up a separation between concepts we will distinguish relationship from link. We use the term link to refer to a path in the graph (described in definition 5). Therefore, we distinguish between two types of links:

- The direct link: is a path in the graph that connects two users that share a common keyword;
- The indirect link: is a path in the graph that connects two users through an existing association of two different keywords, used individually by the users.

The algorithm in figure 3 presents the procedure to find the indirect links for every user, from an existing dataset.

```
Require: $O(U_i) \neq \emptyset$
Require: $T'(U_i)$ from algorithm in figure 2
$I\!K = \{\emptyset\}$
for all $I_t \in O(U_i)$ do
    if $T'(I_t) \neq \emptyset$ then
        for all $K_p \in T'(I_t)$ do
            $I_{K_p} = \{I_r \mid I_r \in I, I_r \notin O(I_r) : K_p \in T'(I_r)\}$
            $I\!K_{K_p} = \{K_r \mid K_r \in K, \forall I_u \in I_{K_p}, \forall I_v \in O(U_i) : K_r \in T'(I_u)$ and $K_r \notin T'(I_v)\}$
            $appendUnique(I\!K, I\!K_{K_p})$
        end for
    end if
end for
return $I\!K$
```

Fig. 3. Frequency analysis experiment. The goal is to obtain all indirect keywords of the user, from the relationships between items and keywords. The $appendUnique(firstList, secondList)$ function appends the second list to the first, avoiding duplicates.

The rank values are determined by the procedure in definition 7. See section VI-A for the experiment's results.

With the indirect keywords identified, it is trivial to associate them with the users, and exclude the user's direct relationships. Fig. 4 illustrates an indirect link (full line), and the direct links (dashed lines) that support it.

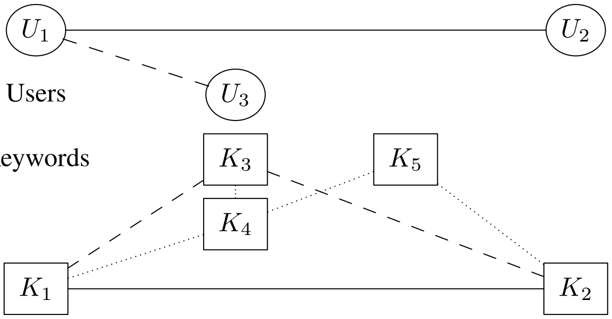

Fig. 4. User U_1 uses keyword K_1, user U_2 uses the keyword K_2, and user U_3 the keyword K_3: the indirect link (full line) between users U_1 and U_2 is supported by the direct links (dashed lines) between keywords K_1 and K_3, and keywords K_3 and K_2. Dotted lines represent other links in the graph.

2) Classification Analysis: Our second method is to train a classifier, which will be able to decide if a keyword is interesting to the user. The classifier is a support vector machine (support vector classifier - SVC), trained with the keywords of the user. The SVC is able to determine if a particular keyword belongs to the user. The success of the classifier is determined by the training conditions, i.e., the set of features used to infer data patterns and the training set. The challenge is in determining if the training set of keywords accurately represents the user's interests, and in selecting a robust set of keyword features.

For a keyword, K_p, removed from the user's (U_i) direct keywords, by the procedure in figure 2, the features are represented by the pair, $\mathcal{F}$, determined by the cardinalities of the feature sets $\mathcal{A}$ and $\mathcal{B}$, such that:

$$\mathcal{F} = \langle |\mathcal{A}|, |\mathcal{B}| \rangle$$

We propose two pairs of feature sets, $\mathcal{F}_a$ and $\mathcal{F}_b$, defined by:

$\mathcal{F}_a$ Each keyword is represented by the number of users that use the keyword and other keywords of the user ($\mathcal{A}$), and the number of keywords that co-occur with it in the user's items ($\mathcal{B}$):

$$\mathcal{A} = \{U_j \mid \forall K_q \in U\!K(U_i) : K_q \in U\!K(U_j)\}$$
$$\mathcal{B} = \{K_q \mid \forall I_t \in O(U_i), K_q \in T(I_t), K_p \in T(I_t)\}$$

$\mathcal{F}_b$ Each keyword is represented by it's absolute number of items ($\mathcal{A}$) and it's absolute number of users ($\mathcal{B}$).

$$\mathcal{A} = \{I_t \mid \forall I_t \in I : K_p \in T(I_t)\}$$
$$\mathcal{B} = \{U_j \mid \forall U_j \in U : K_p \in U\!K(U_j)\}$$

To sort the output of the classifier, we consider two ranking values. The first, R_{K_p}, for every keyword, K_p, with a positive match, is defined by:

$$R_{K_p} = \sum_{K_r \in U\!K(U_i)} |\{K_r \mid \exists I_t \in I : K_r \in T(I_t), K_p \in T(I_t)\}|$$

R_{K_p} calculates the sum of the number of co-occurrences between K_p and the user's keywords. We also adapted the approach from [2], and normalized the ranking values by the frequency of K_p, F_{K_p}, such that:

$$F_{K_p} = \frac{|\{I_t \mid K_p \in T(I_t)\}|}{|I|}$$

$$R'_{K_p} = \frac{R_{K_p}}{F_{K_p}}$$

In section VI-B we report that the second set of features, $\mathcal{F}_b$, and the second ranking method, R'_{K_p} produce better results.

V. Experimental Setup

Our experiments take on the example introduced in section III-A, and execute over data crawled from the Flickr network. In this section we describe the dataset that resulted from the crawl, and the evaluation metrics used for the results' evaluation.

A. Datasets

Flickr provides an API [3] that facilitates querying its content. Through the API, it is trivial to obtain a user characterization from the user name or id. It is also possible to obtain a user's list of photos and one photo's list of keywords. The API also allows the querying of the system for a particular keyword, providing, as a result, the list of photos associated with the keyword.

As it was not possible to completely analyse Flickr's content, one of the tasks of this experiment was to sample the dataset. To meaningfully sample a large social graph we followed the approach of Leskovec, et al. [17], where the authors discuss a set of characteristics present in social networks' graphs, which led to the definition of a graph generation model, the Forest Fire model. This model later inspired a network sampling algorithm [18] which produces a meaningful graph, showing the same characteristics of the original graph. We adapted the sampling algorithm in [18] to deal with the structure of the information present in Flickr. The sample graph represents 912 users, 249 151 items and 116 662 keywords. It contains 2 698 127 edges between items and keywords.

The dataset is distributed in an SQLite database and pickled Python structures. It is freely available at http://img.di.fct.unl.pt/amgs/datasets/.

B. Evaluation Metrics

To evaluate the classification analysis results, we adopted two standard metrics: the mean reciprocal rank and precision at rank. Both metrics are statistics that describe list with a ranking of queries results.

Mean reciprocal rank (MRR): informs where the first relevant keyword occurs in the ranking, averaged over all queries. It is calculated with equation 1.

$$MRR = \frac{1}{N} \sum_{1=1}^{N} \frac{1}{rank_i} \qquad (1)$$

Precision at rank K (P@K): is the proportion of retrieved keywords that is relevant, averaged over all queries. The results of any retrieval method can be divided into the relevant results and the non relevant, and the precision (P) is determined by equation 2.

$$P = \frac{|relevant \cap retrieved|}{|retrieved|} \qquad (2)$$

Precision at a specific rank is interesting because only the top results are ultimately returned to the user. Given that our goal is to present a list of recommendations, which cannot be too long to be effectively delivered by most user interfaces, we show the precision at rank 1, 5, 10, and 20.

VI. Evaluation Results

The evaluation presents two experiments. First, the frequency analysis, that helps us understand the data available in the network and how to use the model to extract information about the network. Second, the classifier training experiment.

The experiments analyze keywords of users. Both experiments execute the procedure in figure 2 for every keyword under analysis. This keyword is referred to as the *removed* keyword. The removed keyword is said to be *recovered* if it belongs to the list of indirect keywords that both experiments output.

A. Frequency Analysis Results

We performed the frequency analysis with the top 20 keywords of each user. The procedure output was:

1) The recovery result for each top keyword;
2) The two ranking scores, i.e., ranking through the number of items and the number of direct keywords (see definition 7, in section IV-A);
3) The general characterization of the keyword, i.e., the total number of items and direct keywords.

The first conclusion drawn from the results was that the ranking method through the number of direct keywords does not produce meaningful results, and we excluded it from further analysis. The remaining analysis is focused on the ranking through the number of items.

Table I shows the keyword frequency analysis over the number of items. We consider two sets of keywords: all the keywords in the dataset, and the set of keywords that were removed, i.e., the top keywords of each user. We see that the average number of items for the keywords removed during the test case is 364.76. We will consider this value while analyzing the recovery results.

TABLE I. Characterization of the number of items associated with a keyword.

Keywords	Mean	Std. Dev.	Mode	Minimum	Maximum	Percentiles		
						25%	50%	75%
All	23.12	191.90	1	1	15513	1	1	5
Test Case	364.76	964.44	1	1	15513	8	67	264

The threshold values used to determine a recovery were estimated after an analysis of the average recovery rate of

[3] The Flickr API can be accessed through urls in the form of ”http://api.flickr.com/services/rest/?method=”, and its documentation is available at www.flickr.com/services/api

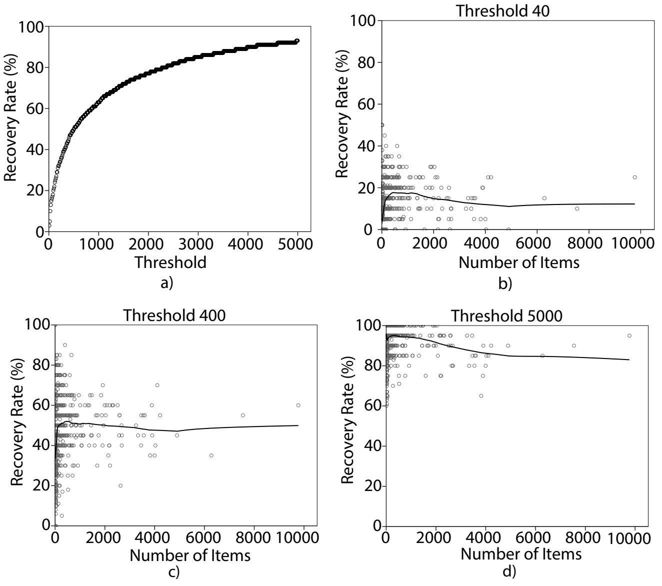

Fig. 5. Frequency analysis results for the users' top 20 keywords' average recovery rates by the user's number of items: a) average recovery rate of thresholds between 10 and 5000, in intervals of 10; b) recovery rate for threshold 40; c) recovery rate for threshold 400; d) recovery rate for threshold 5000. Shows the users' top 20 keywords' average recovery rates by the user's number of items.

several thresholds, between 10 and 5000, with increments of 10. These results are available in Fig. 5.

We now focus on three thresholds: one at 40 (shown in figure 5), which represents the minimum meaningful value, below which there are no useful recovery rates; one at 5000 that analyses the maximum extreme in Fig. 5; and one at 400 that explores values around the average number of items per keyword in the test case (364.76). The average recovery rates (with SD standard deviation and MD mode) are: 12.74 (SD=10.71, MD=0) for threshold 40; 45 (SD=17.45 MD=50) for threshold 400; and 92.6 (SD=8.47, MD=50) for threshold 5000.

The keyword frequency characterization reveals that although the average number of items associated with a keyword is low, these are highly skewed towards much higher values. However, the mode is 1, which means that most keywords are associated with one item.

We found a significant correlation between the recovery rate and the number of items of the user (Spearman correlation coefficient of 0.81, p-value < 0.0005), which is consistent with the lower recovery rate for users with a high number of items, because keywords that are exclusive to the user cannot be recovered by our method - there are no paths in the graph that connect the keyword.

B. Support Vector Classifier Results

The evaluation uses two set of users: a set of 50 users and a set of 300 users. In each we query for 50 keywords for each user. We show the the results for both training sets of features, i.e., $\mathcal{F}_a$ and $\mathcal{F}_b$, described in section IV-B2.

Table II shows the results ranked using the sum of the number of co-occurrences between each keyword and the user's keywords, normalized by the frequency of the keyword, i.e., R'_{K_p}, as described in section IV-B2.

TABLE II. CLASSIFICATION TASK EVALUATION RESULTS.

Number of Queries	Rank	MRR	P@1	P@5	P@10	P@20
50	$\mathcal{F}_a$	0.5064	0.3600	0.3240	0.3080	0.2520
	$\mathcal{F}_b$	0.6372	0.5200	0.3840	0.3140	0.2580
300	$\mathcal{F}_a$	0.2817	0.1800	0.1180	0.0977	0.0783
	$\mathcal{F}_b$	0.3978	0.2667	0.2027	0.1690	0.1355

Although we were not able to reproduce results of keyword or user recommendation methods in the same context as ours, Zhou, et al. [14], present work that is comparable to ours. The main difference is the dataset, which is a crawl of de.licio.us [4], but unfortunately we were not able to obtain. In table III we partially reproduce the authors results, and compare them with our best method, $\mathcal{F}_b$ + R'_{K_p}, where we obtain an improvement in the mean reciprocal rank.

TABLE III. COMPARISON WITH ZHOU, ET AL. [14]

Method	MRR	P@R
$\mathcal{F}_b$	0.3978	0.2667
Zhou, et al. [14]	0.2345	0.3272

C. Future Case Studies

Our case study uses a network that does not impose strict rules for keyword creation. Flickr adopts a very permissive strategy for annotating items: the users can use any term, which, upon creation, becomes globally available on the network. When annotating photos, some users reuse popular keywords while others use their exclusive keywords.

Another important aspect of Flickr is that authorship of photos is not collaborative, and only one user decides when annotating an item, without need for agreement with anyone else. These are consequences of not imposing a classification strategy, which would require the use of general taxonomies. This may not be practical for Flickr, but is suitable for other types of networks.

The definition of the model led to the construction of a visualization and interaction component, which provides a visual context to indirect relationships. It uses the model's elements interchangeably as node or edge, to change between different representations of the information. Fig. 6 shows examples of graph visualizations using our case study dataset. We plan to explore the potential of the component with the future case studies.

We plan to test our method with a sample of the IEEE Xplore Digital Library [5]. It has a keyword taxonomy that most articles follow. Some articles also have author generated keywords, but the average number of keywords per article is considerably smaller comparing with Flickr. We also include a case study with the Arxiv [6] library, which is not as strict as the IEEE Xplore

[4] www.delicious.com

[5] http://ieeexplore.ieee.org

[6] http://www.arxiv.org

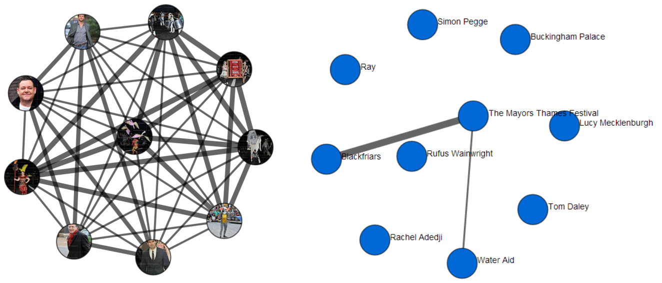

Fig. 6. Flickr's case study dataset visualization. On the left are several users, connected by common keywords. On the right are several keywords, connected by common items. Note that the thickness of the connection indicates the rank value.

VII. Conclusions and Future Work

This work presented an approach to the problem of identifying meaningful suggestions on a productive community, based on the structure of the content generation network. To address the challenge, this paper presents three contributions:

1) A model to represent productive networks – see section IV-A;
2) An indirect keyword discovery method to build a recommendation list – see sections IV-B1 and IV-B2;
3) An indirect keyword ranking method to sort the list – see sections IV-B1 and IV-B2.

We developed a model to represent the collaborative potential of a productive community, which defines a graph representing a community's information structure, through the use of three basic concepts: user, item and keyword. We also explored methods to find potentially interesting keywords and user relationships.

The model was evaluated by two experiments, using a dataset built from the Flickr network. The starting point was the removal of a keyword from the user' items, and the common goal of the experiments was the recovery of that keyword as an indirect keyword. We believe that this method enables the conclusion that indirect keywords are relevant to the user. The experiments were a frequency analysis and a classification analysis. Both produced relevant results.

The frequency analysis, which consisted on the application of the procedure for finding implicit potential relationships, described in section IV-B1, concluded that it is more difficult to build suggestions to users with a high number of items. The classification analysis, described in section IV-B2, produced results that improve on comparable methods.

We were also able to identify the effects that a non-restrictive keyword policy has on the usefulness of keywords for indirect keyword identification, which motivated the outline of future case studies on communities with different annotation policies.

References

[1] G. Smith, *Tagging: people-powered metadata for the social web*. New Riders, 2008.

[2] B. Sigurbjörnsson and R. van Zwol, "Flickr tag recommendation based on collective knowledge," *Proceeding of the 17th international conference on World Wide Web - WWW '08*, p. 327, Apr. 2008.

[3] C. V. Damme, M. Hepp, and K. Siorpaes, "Folksontology: An integrated approach for turning folksonomies into ontologies," *Bridging the Gep between Semantic Web and Web 2.0 SemNet*, 2007.

[4] H. Xu, X. Zhou, M. Wang, Y. Xiang, and B. Shi, "Exploring Flickr's related tags for semantic annotation of web images," *Proceeding of the ACM International Conference on Image and Video Retrieval - CIVR '09*, p. 1, 2009.

[5] M. Roth, A. Ben-David, and D. Deutscher, "Suggesting friends using the implicit social graph," *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 233–241, 2010.

[6] M. Heckner, T. Neubauer, and C. Wolff, "Tree, funny, to_read, google: what are tags supposed to achieve? a comparative analysis of user keywords for different digital resource types," *Proceedings of the 2008 ACM workshop on Search in social media*, pp. 3–10, 2008.

[7] O. Nov, M. Naaman, and C. Ye, "What drives content tagging: the case of photos on Flickr," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1097–1100, 2008.

[8] N. Garg and I. Weber, "Personalized, interactive tag suggestion for flickr," *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08*, p. 67, Oct. 2008.

[9] H. Liang, Y. Xu, Y. Li, R. Nayak, and X. Tao, "Connecting users and items with weighted tags for personalized item recommendations," in *Proceedings of the 21st ACM conference on Hypertext and hypermedia - HT '10*. New York, New York, USA: ACM Press, Jun. 2010, p. 51.

[10] T. Lappas, K. Punera, and T. Sarlos, "Mining tags using social endorsement networks," *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*, p. 195, 2011.

[11] K. Stefanidis, N. Shabib, K. Nø rvåg, and J. Krogstie, "Contextual recommendations for groups," *Advances in Conceptual Modeling: ER 2012 Workshops*, 2012.

[12] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang, "Tag ranking," *Proceedings of the 18th international conference on World wide web - WWW '09*, p. 351, 2009.

[13] Z. Wang, J. Feng, C. Zhang, and S. Yan, "Learning to rank tags," *Proceedings of the ACM International Conference on Image and Video Retrieval - CIVR '10*, p. 42, 2010.

[14] T. C. Zhou, H. Ma, M. R. Lyu, and I. King, "UserRec: A User Recommendation Framework in Social Tagging Systems." in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010, pp. 1486–1491.

[15] E. H. E. Chi and T. Mytkowicz, "Understanding the efficiency of social tagging systems using information theory," *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pp. 81–88, Jun. 2008.

[16] A. Zubiaga, C. Körner, and M. Strohmaier, "Tags vs shelves: from social tagging to social classification," *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pp. 93–102, Jun. 2011.

[17] J. Leskovec, J. Kleinberg, C. Faloutsos, H. D. Management, and D. Applications, "Graphs over time: densification laws , shrinking diameters and possible explanations," in *Proceeding of the 11th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, New York, USA: ACM Press, Aug. 2005, pp. 177–187.

[18] J. Leskovec and C. Faloutsos, "Sampling from large graphs," *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.