

# Evolving Social Search Based on Bookmarks and Status Messages from Social Networks

Bastian Karweg  
Mobile Advertising GmbH  
Karlsruhe, Germany

Christian Hütter  
Karlsruhe Institute of Technology  
Karlsruhe, Germany

Klemens Böhm  
Karlsruhe Institute of Technology  
Karlsruhe, Germany

Karweg@mobile-advertising.com

Christian.Huetter@kit.edu

Klemens.Boehm@kit.edu

## ABSTRACT

*Social search* is a variant of information retrieval where a document or website is considered relevant if individuals from the searcher's social network have interacted with it. Our ranking metric *Social Relevance Score (SRS)* is based on two factors. First, the *engagement intensity* quantifies the effort a user has made during an interaction. Second, users can assign a *trust score* to each person from their social network, which is then refined using social network analysis. We have tested our hypotheses with our search engine [www.social-search.com](http://www.social-search.com), which extends the existing social bookmarking platform [folkd.com](http://folkd.com). Our search engine integrates information the [folkd.com](http://folkd.com) users share through the popular social networks Twitter and Facebook. With permission of 2,385 testers, we have connected to their social graphs to generate a large-scale real-world dataset. Over the course of a two-month field study, 468,889 individuals have generated 24,854,281 website recommendations. We have used those links to enhance their search results while measuring the impact on the search behavior. We have found that social results are available for most queries and usually lead to more satisfying results.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *Information filtering, Search process.*

## General Terms

Algorithms, Performance, Design, Experimentation.

## Keywords

Social Search, Social Networks, Social Graph, Search Engine, Status Messages, Friend Networks.

## 1. INTRODUCTION

### 1.1 Motivation

With the huge growth of information available on the internet, it is increasingly challenging to pick the 10 first results for a query. According to [14], combining classical approaches such as measuring the number of links that point to a website (e.g., PageRank) with complementary ranking criteria from the user's social network increases result quality significantly. Considering

broad topics like “books” or unspecific queries like “restaurant”, leveraging information about the searcher can lead to better results [10]. Consider the following use case:

### Example 1: The pancake recipe

*Jane is looking for a pancake recipe which one of her friends has recommended. Searching on Google for 'pancake recipe', she only has a slim chance of finding it. Using social search, Jane is more likely to get results that both contain the query and have been recommended, commented upon, bookmarked, or liked by her friends.*

Research has proposed various approaches to social search, e.g., [2],[4],[6]-[9]. However, we are not aware of a systematical evaluation of how users interact with real data from their social networks. One of the authors runs a startup company that provides the social bookmarking platform [folkd.com](http://folkd.com). This service has more than 1,500,000 users worldwide who share over 24,000,000 bookmarks. We leverage this massive user base to evaluate social search techniques that are promising in a real-world setting. More specifically, we examine the following questions: (a) Do users generate enough content such as bookmarks and status messages to serve as data base for an internet search engine? (b) How many friends does an average user need to obtain good search results tailored to him personally? (c) Do complementary ranking criteria from the user's social network actually improve search quality? – Studying these questions is difficult because one needs a very large number of users who grant access to their social networks. Furthermore, it is still an open issue how to carry out social search such that users get satisfying results on real-world data.

[Folkd.com](http://Folkd.com) already comes with a full-text search engine that allows users to search through the bookmarks. We have extended this search engine to rank the result elements of a query according to their social relevance for the user. We refer to the metric that we use as *social relevance score (SRS)*. It calculates the result ranking, individually for each user, based on two factors, which we call *engagement intensity* and *social trust*. The *engagement intensity* specifies how strongly a user has interacted with a result element using social services. For example, both clicking on a link and recommending a restaurant to a friend are interactions/engagements. However, the first one is weaker than the second one. The *social trust* determines how intense the relation of the user is to other individuals from his social network. This value is fine-tuned based on the reputation of the users in the network. Apparently, some users are more active than others, and not all individuals from the user's social network are equally trustworthy. One hypothesis, which we examine in this paper, is that the ranking improves as the searcher adds more friends, or his friends create more content over time.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

Copyright 2011 ACM 978-1-4503-0717-8/11/10...\$10.00.

Our social search engine [www.social-search.com](http://www.social-search.com) is operational since January 2011. Users of [folkd.com](http://folkd.com) may connect their accounts to Facebook and Twitter. These networks currently are popular to publish links and recommendations for friends through what is called “status messages”. Being allowed to aggregate status messages of these users as well as their friends, we have conducted a field study that has run for 58 days and has had 2,385 participants. The participants had more than 460,000 friends, who have published 24,854,281 link recommendations during that time. After filtering for spam, empty and duplicate queries, we have been able to collect (with permission of the users) 2,098 sessions to analyze.

On average, each friend creates one new link recommendation per day. The number of search results available also depends on the social graph of the users, the activity of their friends, and the broadness of the search term. Given a network of 50 friends, the average user will get about one highly relevant social result for a random query. If he has as many as 300 friends, social search even delivers more than 5 social results on average. The results of our field study show that, using [social-search.com](http://social-search.com) with SRS ranking, users find a satisfying result about 2 to 3 positions earlier and with 0.25 clicks less than using plain [folkd.com](http://folkd.com) with a classic full text ranking. We conclude that ranking search results through social criteria can lead to great improvement in search quality.

## 2. RELATED WORK

According to [6], there are three types of social search: collective, collaborative, and friend-filtered. Collective search leverages the “wisdom of the crowds” [13] to assess the relevance of a website. Collaborative search is based on the “village paradigm” [2] where experts from the social network of the user answer search queries. Friend-filtered search offers personalized results based on recommendations from the user’s social network. Our work adapts the friend-filtered approach. We assign users to different levels of trust based on their social relationship. The user may adjust these levels to control how much which friend is influencing his personal rankings.

Research has proposed various metrics for ranking social search results. *SBRank* [4] simply counts how many users have bookmarked an item. It focuses on collective social search and does not respect different engagement types or different trust levels. More closely related to this study are *SocialTrust-Ensemble* [12] and *FriendshipSimilarity* [1], which do incorporate friend recommendations. However, these measures lack the option of individual adjustments. In addition, both require the system to precalculate the results on a static data base. This renders their application difficult on systems whose content changes frequently, like social web services.

Next, according to the classification in [16], there are three types of queries: navigational, informational, and transactional. With navigational queries, the user already knows the destination he wants to reach, e.g., “CIKM 2011”. Most queries are informational, where the user searches for a certain topic, but does not know the destination. Transactional queries always reveal the next intended action of the user, e.g., buying a certain product after searching for “cheap digital camera”. We expect navigational queries to benefit the least from social search. In contrast, we hypothesize that transactional and informational queries in particular can capitalize on friend recommendations.

Some approaches to social search [7], [13] rely on tags used to represent a topic in a folksonomy. However, a single tag can hardly cover an entire topic and is more ambiguous than a

contextual sentence [8]. Given the recent developments in social media, the usage of tags is on a decline [9]. People rather use status messages to describe a link. This contextual information serves as a personal recommendation and is more trustworthy than metadata provided by the content producers [4]. Further, [8] sees status messages as “good summaries” of the content and as a good approach to better understand why something has been recommended.

Our metrics to classify user relations have been inspired by [1] who have experimented with a similar value called *FriendshipStrength* and [12], who have proposed a probabilistic matrix-factorization framework to combine characteristics of user recommendations by his friends. Their metric did not include flexible trust levels, but tried to estimate the strength of a relation by deriving transitive relations (e.g., same interest in a certain topic). However, both have followed the assumption that “you trust recommendations of your close friends more than those of your casual acquaintances”.

## 3. OVERVIEW

When social bookmarking services like [folkd.com](http://folkd.com) or [digg.com](http://digg.com) started to implement friend connections and recommendations, users did not adapt it very well. They instead already had opted for the opposite, using bookmarking or sharing functionalities that social networks like Facebook or Twitter had implemented. The demand for a unified and easy access to this information grows steadily. With a massive amount of nearly 7,000 tweets<sup>1</sup> and 16,000 status messages<sup>2</sup> per second it is justified to assume that these messages contain a lot of relevant links that can improve search results. We leverage the user base of [folkd.com](http://folkd.com) to aggregate link recommendations from social networks. We have implemented the search engine [www.social-search.com](http://www.social-search.com) that takes these link recommendations into account.

### 3.1 Data Aggregation

Before discussing the search engine in detail, we describe what happens when a new user joins our platform. We have recruited participants by means of the well-known pyramid scheme: we have invited randomly selected users from [folkd.com](http://folkd.com). They then announced their participation through their social network accounts, which in return lead to a chain reaction of more friends joining. This has worked very well, and the group of participants is still expanding even after the actual experiment has finished.

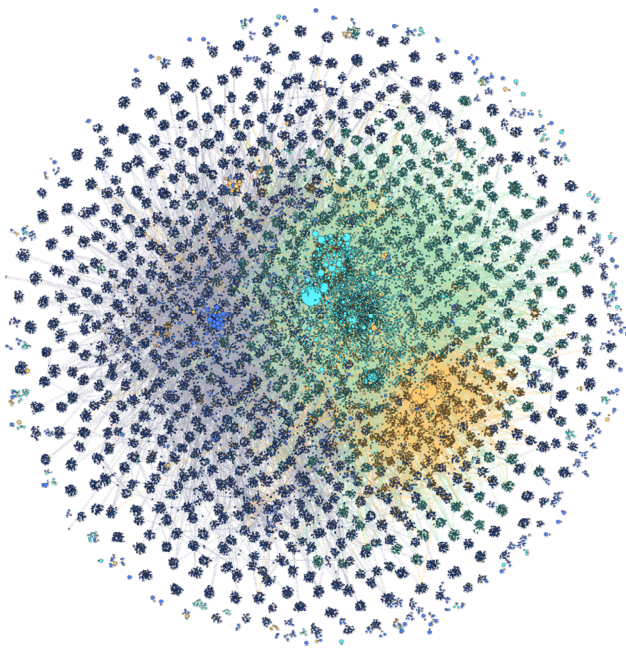
The first step when a user connects his [folkd.com](http://folkd.com) account with a social network is to aggregate the profile data from the account of the user and his network of friends. All of the platforms discussed above provide APIs for external developers. Our system parses the social content streams to determine if they contain a link reference to a web page. Aside from simple links, the Facebook API also provides references to other “objects” in the so-called social graph. For example, if somebody “likes” a band or a celebrity, the network creates a “fan” relation. We combine these relations and links into what we call an *interaction* or *user engagement*.

As a side note, we mention that there is a very common technique to shorten links that are posted using services like [bit.ly](http://bit.ly) or [tinyurl](http://tinyurl).

---

<sup>1</sup><http://blog.Twitter.com/2011/01/celebrating-new-year-with-new-tweet.html>

<sup>2</sup><http://www.Facebook.com/press/info.php?statistics>





**Figure 1. Visualization of User Relations in Test Dataset**


It is necessary to expand those to prohibit the creation of duplicate documents. Our crawler fetches the meta tags of the web page, checks for dead links and sorts out spam.

In the third step, we replicate the user-to-user relations existing in the social networks. To give an idea of how the users are connect to each other we have visualized a selection of 40,000 users (nodes) and 65,000 friend relations (edges) in Figure 1. This sample consists of all participants of our field study (see Section 5) plus those users who could be reached by following one hop of inbound or outbound friendship relations. We only considered additional relations from those friends if they reached a user that has already been selected. The total number of users in Figure 1 is limited to 100,000 due to limitations of the visualization tool. Looking at the edges of the graph, the small circles of flower-shaped groups correspond to testers whose friends do not have any other inward connections to the graph. The visualization was generated by using Gephi [18]. The layout of the graph was produced using the OpenOrd algorithm. The network diameter here is 8 with an average degree of 2.8.

The size of the nodes is in line with their PageRank [5]. The layout algorithm tends to locate nodes with a higher degree more in the middle. An interesting effect shows when aligning the nodes according to their source: While some accounts already existed in folkd.com (orange in bottom right), others have been

#### Task Management At Its Best With wunderlist | 6Wunderkinder

 **Bastian**,  **polarity**: Wer ne Desktop App für Todos sucht, kann sich mal "Wunderlist" anschauen LINK ... 6 Wunderkinder - Does using your software feel like rocket science? We think it's time for a change!

 **pmoehring**: beautiful task management, but hey, still task management :) RT @bibrue: cool stuff. LINK ... 6 Wunderkinder - Does using your software feel like rocket science? We think it's time for a change!

[www.6wunderkinder.com/wunderlist](http://www.6wunderkinder.com/wunderlist)

**Figure 2. Example of a Social Search Result**

created when pulling data from Twitter (light top right) and Facebook (dark on the left). This results in clusters of people of the same group. Other individuals who are connected to multiple networks are located between these clusters. We also could observe that the connectivity in the Twitter network is higher. This is because it does not enforce a confirmation when creating a connection.

A very important issue for us has been to respect privacy settings when including data from Facebook or Twitter. If a link was posted as public, we could use it freely. If it was posted to be only visible to friends, we have used this link for the social search of these friends, but not for other users. We refer to [19] for a discussion of more sophisticated privacy settings.

The total time for initializing and calculating the user's first individual social search result depends on the amount of data that has to be gathered. We set up a periodical re-crawl (every 15 minutes in the background) to fetch new or updated data for the user.

## 3.2 Social Search

Once the data is aggregated (which currently takes about 30 seconds), the user's first query is ready to be executed. The effect of our SRS metric is that results his friends have interacted with are pushed to the top. Figure 2 shows an example of a search result with our implementation. We point out that each entry also shows the name of the user (preferably a friend) who has engaged with the result item as well as the comment of that user.

The user is also able to refine the search to be more objective or more social. This adjustment allows for numerous applications: The most basic one is that users search for information that friends have recently mentioned, cf. *Example 1*. Another application is that a search for "YouTube" not only displays a link to the popular video site but also shows videos that friends have enjoyed most.

Interestingly, social search has also caused the inverse behavior: Testers started actively seeking out people to befriend who had served good recommendations around certain topics like photography, celebrities and others. Our approach of letting users arrange their friends into different trust levels supports this (see Section 4.1).

## 3.3 Technology

With our system all data is stored in a replicated, relational database. To support fast queries on the huge dataset we have indexed important tables using high-performance full text indexes built with SphinxSearch<sup>3</sup> in addition. The application itself is handling the search requests from the user as well as constantly updating the latest available social data by accessing the external social graph data through the APIs of Facebook and Twitter. As core programming language for this business logic PHP5 with a custom made MVC-framework was used. To calculate the individual user trust levels we needed to calculate different centrality measures on our user relations table, which we found to be most easily done by MATLAB. The presentation tier is a classic web-frontend application, running on top of an Apache2 web server executing simple HTML5 and AJAX components.

The size of the search index in our project has become approximately 0.5 gigabyte per one million documents indexed.

<sup>3</sup><http://www.sphinxsearch.com>

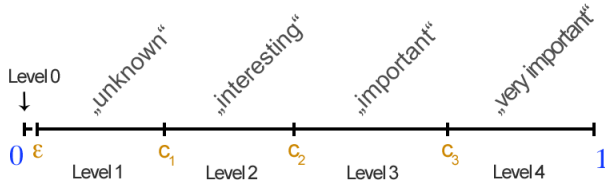


Figure 3. Different Trust Levels

But even with a fast and out-of-the box full text solution like SphinxSearch the high dynamics of internet and social media content is challenging for the system. The frequent (near real-time) document updates in particular are problematic. Indexing software is adding additional documents fast, but frequent updates of existing ones will slow down query times significantly. From a user point of view updates need to be fast. When a friend posts a new link, the user wants to be instantly able to find it, also if that specific links had also been posted by somebody else in the past. We have used parallel indexes in combination with a cached Delta Index to provide live updates as well as very fast querying.

### 3.4 Full Text Relevance

In the regular folkd.com system, users can search through bookmarks that users have added manually. The system ranks them by a full text search metric which is calculated upon the titles, descriptions and tags of the web paged recommended by the users. We combine two approaches when calculating the full text score: phrase ranking and statistical ranking. Phrase rank is based on the length of the longest common subsequence (LCS) of the search term and the documents, the statistical approach is a based on the probabilistic BM25 retrieval framework introduced by [17] which considers word frequencies.

For the implementation of folkd.com, we rely on the full text search engine SphinxSearch<sup>4</sup>. They state that, while in general better partial phrase matches are ranked higher, one can achieve better full text results by pushing perfect phrase matches to the top. Having tested different metrics on the folkd.com system, we have observed that, considering full text rankings, plain BM25 indeed performs better on single-word queries. However, the combination with the phrase proximity based ranking provides a better search quality overall than any statistical scheme alone.

## 4. RANKING THE SEARCH RESULTS

In our effort to improve the search experience for our users, we identify commonalities of existing approaches to social search, which we then combine into a new ranking metric called *Social Relevance Score* (SRS). We also say how we have combined the new ranking with classic full text ranking metrics.

### 4.1 Trust Score

Analyzing the dynamics in social networks, [3] and [6] have found similar usage patterns: Some users are very active, sharing a lot of information with their friends, whereas others just follow what is going on in their social circle. Some social networking services such as Facebook offer the possibility to arrange friends into groups. However, this grouping does not have any

implication on how the service handles link recommendations of individuals from different groups. In the worst case, all links published by any friend are contained in one update stream. This causes the problem that the update stream tends to be dominated by few users. Consider the following scenario:

#### Example 2: Lost in the shuffle

*Kate is interested in volleyball; therefore she is listed as a “fan” of various volleyball teams on Facebook. Most of the teams regularly publish their match scores. By searching for “volleyball”, Kate could easily overlook the personal invitation of a friend to a volleyball event. By not being able to distinguish between different levels of trust, this recommendation by a close friend can easily get lost in the shuffle.*

To address the issue discussed in Example 2, we propose to have five trust levels, which the system assigns to each friend. The user can then adjust these by hand. If a Person *A* recommends a link, and *A* is a friend of the searcher, then the trust level of *A* influences the personalized ranking of that link. The more important *A* is to the searcher (i.e., the higher his trust level), the higher the link will be ranked. If the trust between the two persons is zero (i.e., the lowest level), then the recommendation by *A* has no influence at all.

The *trust levels* follow the schema presented in Figure 3. Our search engine individually calculates the trust levels of all individuals based in their relationship to the user. The user can then adjust the trust level for each of his friends by hand. The 0-value is the lowest level used to block others intentionally. Level 1 is the default setting for unknown users. Level 2 is intended for contacts that are loose but interesting, e.g., organizations or celebrities the user is a fan of. Level 3 should be used for confirmed relationships like real friends on Facebook, LinkedIn, or Xing. Finally, one may promote special relations like close friends or family members to Level 4. Note that the assignment to a certain level is not reflexive and remains undisclosed to the user affected. There are various ways to calculate the borders  $c_i$  of each trust level  $i$ . We preferred an inverse logarithmic scale along the number of people who fall into those clusters.

The *trust score*  $t_s(x)$  of user  $s$  for his friend  $x$  ranges between the lower and the upper bound of  $x$ 's trust level  $c_s(x)$ . The system adjusts the trust score within this level according to the overall popularity  $r(x)$  of  $x$  in the social network.

$$t_s(x) = c_s(x) + r(x)$$

That is, popular friends have more impact. This is because the popularity  $r(x)$  puts their trust score within the bounds of the trust level assigned to them. To determine the popularity of each user, we calculate the PageRank [5] centrality measure on the graph consisting of the friend connections from the social network. We have chosen PageRank because it combines good accuracy with moderate computation complexity.

### 4.2 Engagement Intensity

In the time before web 2.0, user interactions were limited to the creation of links from one website to another one [4]. Nowadays, there are various online interaction possibilities. We propose the term *user engagement* as a generic term for these possibilities, standing for all cases where a URI identifies the recommending user and the object of interaction. In other words, whenever one can track User *X* (link to his user profile) having interacted with

<sup>4</sup> <http://sphinxsearch.com/docs/current.html#weighting>



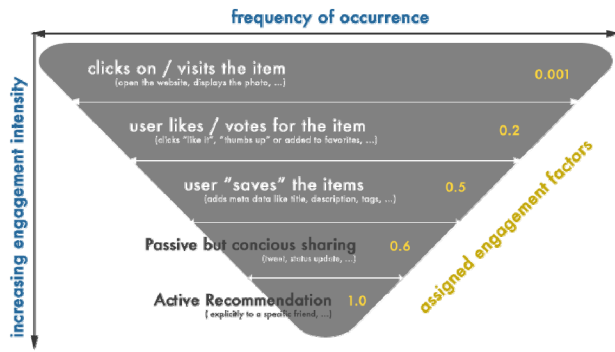


Figure 4. Different Levels of User Engagement

Object  $Y$  (link to their profile/homepage), we have a user engagement. Examples are restaurants reviews, recommending music and rating books or movies [8]. All these engagements express relevance, and a social search engine should consider them. Status messages on Facebook and Twitter have become widespread aggregators for those recommendations. This is because many services have integrated their interfaces.

Next, not all interactions are equally intense: The engagement factor should increase with the effort the user spends in order to engage in this specific way. For example, giving a “thumbs up” to a movie trailer is much easier than writing a review about a restaurant. This is why we have come up with the notion of *engagement intensity*  $e_x(i)$ , i.e., how much user  $x$  has engaged with document  $i$ . The scale ranges from 0 (neutral, no engagement) to 1 (strong engagement). Figure 4 shows the different levels of engagement.

Note that the range of the engagement-level values is continuous. This allows to reflect how often different engagements occur in relation to each other. For example for about every 200 visits, one vote has occurred. Spreading the levels relative to each other across the scale, the lowest level has a factor of 0.001, the second one of 0.2 etc. In case of several engagements of a user with a resource (e.g., he both clicks on it and rates it), the highest engagement factor counts.

### 4.3 Social Relevance

Our new *Social Relevance Score* (SRS) ranks the result elements of a query according to their social relevance for the user. The ranking considers how intense the interactions with the result items have been, and how much the searcher trusts the individuals recommending a link.

$$SRS_s(i) = \sum_{x \in E_i} (t_s(x) \cdot e_x(i))$$

The SRS is calculated for each element  $i$  of the result set of a full text search (see Section 4.5). For each user  $x$  who has interacted with (i.e., recommended) a document  $i$ , we add up the product of the trust score  $t_s(x)$  of searcher  $s$  in that user  $x$  and the engagement intensity  $e_x(i)$ .

Note that the engagement of several low-trust users can outweigh a recommendation of a close friend. Further, the popularity of the user interacting in the network influences his trust score. This means that users who are important to the social network gain more influence compared to others on the same trust level.

Figure 3 shows that we define the lower border of Level 1 by a value  $\epsilon$  above zero. By setting the value of  $\epsilon$ , we define how big the impact of an average unknown user on SRS should be. For example, a YouTube video with one million views can gain a high SRS value even if none of the friends of the user has seen it yet. Increasing  $\epsilon$  boosts documents that are popular in general.

### 4.4 Combining SRS with Full Text Relevance

Considering the different types of search queries, it sometimes makes sense to deliver objective results, while sometimes the searcher is looking for results biased by recommendations from his social contacts. To address this issue the *combined social rank* (CSR) is composed of the classical full text relevance (FTR) of a document and the social relevance score (SRS). Parameter  $\alpha$  serves as a weight:

$$CSR(i) = \alpha \cdot SRS(i) + (1 - \alpha) \cdot FTR(i)$$

Because it is difficult and sometimes impossible to automatically classify queries by type, we have made  $\alpha$  adjustable by the user during the search, using a slider. Whenever the user wants more objective results or more socially biased ones, he can simply adjust the slider and repeat the search.

To save the necessity of computing an SRS value for all elements that contain a certain word, we first send each query to a full text search engine. We then only calculate SRS for the top 1,000 FTR results.

A search for the empty word “” turns off the full text relevance. This transforms social search into a plain recommender system of elements friends have interacted with in the past.

Search quality is boosted by the aggregation of user-generated content that relates to a given web page. The following example shows how three Twitter messages (“tweets”) about a website can add valuable context data:

#### Example 3: Status messages

While bookmarks of a page are structured:

ID: #453567  
URL: http://www.flickr.com  
Title: Welcome to flickr - Photo Sharing  
Tags: photos, flickr, photography, photo, sharing, images

Twitter and Facebook status messages embed a link into a contextual sentence:

@thomas56: “using [flickr.com](http://www.flickr.com) to store and share our family vacation photos”  
@photographerX: “interested in my photography work? see my portfolio on [flickr.com](http://www.flickr.com)”  
@Areporter: “I just uploaded fotos of today’s train crash at grand central to [flickr.com](http://www.flickr.com)”

Example 3 illustrates the difference between placing a link into a natural context instead of performing a manual classification. This leads to broader contextual document and better search results: For example, different spellings like “fotos” and “photos” become searchable.

## 5. EVALUATION

To evaluate our social search engine in a real-world setting, we have conducted an extensive field study. We have formulated various hypotheses, referring to different aspects of our approach.

We then discuss the evaluation criteria we have used to test our hypotheses.

## 5.1 Hypotheses

Comparing the number of links bookmarked on folkd.com with the number of links posted to Facebook and Twitter, we notice an increasing dominance of sharing through social networks over classic methods of content exchange like shared bookmarks. We hypothesize that social networks already produce enough content to fuel an internet social search engine:

### ▪ Hypothesis 1: Link recommendations

Status messages of social networks produce a significant number of link recommendations usable for social search.

Following the observations of [3], [4], and [13], the participation of users in social bookmarking services can be described using a power law distribution. That is, someone with a few active friends will benefit more from the service than someone who is connected to many people who are inactive. We expect that this observation also holds for the amount of link recommendations that are produced for social search.

### ▪ Hypothesis 2: Social graph

The number of links available for social search depends on how active the friends of the user are.

A result of related work is that “a searcher can improve search by joining larger networks and having more friends” [11]. However, we are not aware of any study that provides concrete information on how many friends a user needs for social search to work. We hypothesize that there is a minimum degree of connectivity for social search to work.

### ▪ Hypothesis 3: Search quality

There is a minimum number of friends the user needs to be connected to for social search to produce an adequate number of results.

From [2] we know that the more complex a query is, the longer its search term tends to be. For systems based on a full text search, we assume that increased complexity will lead a smaller result base to filter and re-rank according to social factors. The consequence will be fewer results of the social search.

### ▪ Hypothesis 4: Query types

The number of results depends on the type (informational, navigational, and transactional) and length of the query.

Even when a user with a reasonable number of friends has submitted a suitable query, the most important question remains: Does data from social networks improve search quality? According to [11], results from social networks “provide richer and more reliable clues about the purposes and interests of the user”. We therefore assume:

### ▪ Hypothesis 5: Ranking improvement

The result ranking of classic full text search improves when combining it with the Social Relevance Score (SRS).

## 5.2 Evaluation Metrics

We now explain the metrics used to test our hypotheses. In classic search-engine architectures, the crawler clearly finds enough links on a website to follow. This is not necessarily true for status messages in social networks because their availability depends on

the friends of the user. A first metric to test Hypotheses 1 and 2 is the correlation between the number of friends the user has and the number of links they produce in a given timeframe.

The classic measures of search quality are precision and recall [15]. For social search however, it is difficult to decide whether a document is relevant for a query. This is because relevance might be different for each user. So there is not only one reference set but many different ones. In addition, the high dynamics of the data set can cause the pool of documents noticeable by the individuals to change, i.e., when a user adds or removes friends from his network. A document is noticeable in social search if it contains the query term at least once (i.e.,  $FTR > 0$ ) and at the same time has an engagement of at least one of the user’s friends (i.e.,  $SRS > 0$ ). To test Hypothesis 3, we count how many results social search generates compared to full text search. That is, we measure how the result set changes by incorporating user recommendations.

Another issue is the length and type of queries entered by a user (Hypothesis 4). To evaluate it, we have looked at the word count per query as well as at their type (informational, navigational, and transactional).

Finally, the metrics we have used to measure the improvement of the ranking (Hypothesis 5) are:

- the *RankingPosition* of an item that the user has clicked,
- the *ResultClickCount*, the number of results the user has clicked, as well as
- the *SearchTime*, the latency between the search and the first click on a result item.

We assume that by clicking on a specific result one expresses his interest in the information behind it. Consequently, if the user clicks on a higher result position (position 1 is the highest), this indicates a better ranking. A good search result also requires fewer clicks per query. Namely, the user has been able to find a suitable result with less effort. Finally, taking longer to decide which site to visit indicates that the result tends to be worse/less clear.

Letting the user adjust his search to be more objective or more biased by social engagements, we also have obtained good indications about when either ranking (SRS or FTR) could improve a certain result. Thus, we set the parameter  $\alpha$  to be 0.5 initially. If the users repeat a search request with a higher  $\alpha$  value, we conclude that social recommendations could improve the results shown. In what follows, we will also look at the adjustment of  $\alpha$  in repeated searches.

## 6. Experimental Results

In this section, we describe the results of our field study.

### 6.1 Analyzing Data from Social Networks

Our field study has lasted for 58 days, from November 9, 2010, until January 5, 2011. Table 1 contains some descriptive statistics on the data collected during our field study.

Averaging over all testers, we have found that each user produced one link recommendation per day on average. This observation is independent of the number of friends and their level of activity. Multiplied by the average number of friends, each user receives  $130 \cdot 30 \cdot 1 = 3,900$  new links a month. This probably is too little data to fuel a search engine. If we also include friends-of-friends, we get  $130^2 \cdot 30 \cdot 1 = 507,000$  new links a month (not counting overlaps). This in turn appears to be a decent corpus size for a personalized social search engine.

**Table 1. Social Stream Dataset**

Number of testers	2,385	
Number of friends of testers	468,889	
Number of links crawled	24,854,281 = 428,522 / day	
Data by Social Network	Facebook	Twitter
Number of testers	1,651	1,164
	430 people using both networks	
Number of friends	217,010 (46.28%)	251,879 (53.72%)
AVG No. of friends per user	131.44 ( $\pm$ 281.80)	216.40 ( $\pm$ 410.25)
No. of links crawled	3,344,286	21,509,995
No. of links per user-stream	2,026 = 34.92/day	18,480=318.62/day
No. of links per user-friend	18.70 ( $\pm$ 175.07) = 0.33/day	102.61 ( $\pm$ 600.14) = 1.77/day

We also have tested for a correlation of the number of link recommendations a user receives and the number of friends he has. The data shows a strong positive correlation ( $r=0.58$ ,  $p < 0.05$ ), which supports Hypothesis 2. In summary:

**Result 1:**

- Having more friends will provide more content for social search.
- On average, users get one new link recommendation per friend per day.
- To get a significant number of link recommendations from social networks, also friends-of-friends must be considered.

Somebody with fewer friends may have more results if he is connected to more active people (Hypothesis 2). To test this, we have analyzed the distribution of links shared through the

networks Facebook and Twitter. Applying the Kolmogorov-Smirnov test shows that it actually is a power law distribution, returning estimated fit parameters of [ $\alpha = 2.4100$ ,  $x_{\min} = 133$ ,  $L = -2.8150e+04$ ,  $p < 0.05$ ,  $\text{gof} = 0.0178$ ] for Facebook and [ $\alpha = 2.1600$ ,  $x_{\min} = 587$ ,  $L = -6.4030e+04$ ,  $p < 0.05$ ,  $\text{gof} = 0.0314$ ] for Twitter users.

Moreover, the statistics reveal that the average number of links shared on Twitter is about twice the number on Facebook at the lower activity levels. In fact, we have found a lot of examples such as in Table 2 where a user had more results despite having fewer friends. This heavily depends on the search topic. If one chooses to befriend someone with great knowledge regarding a certain topic this would obviously benefit his queries in that field.

**Table 2. Social Results comparison**

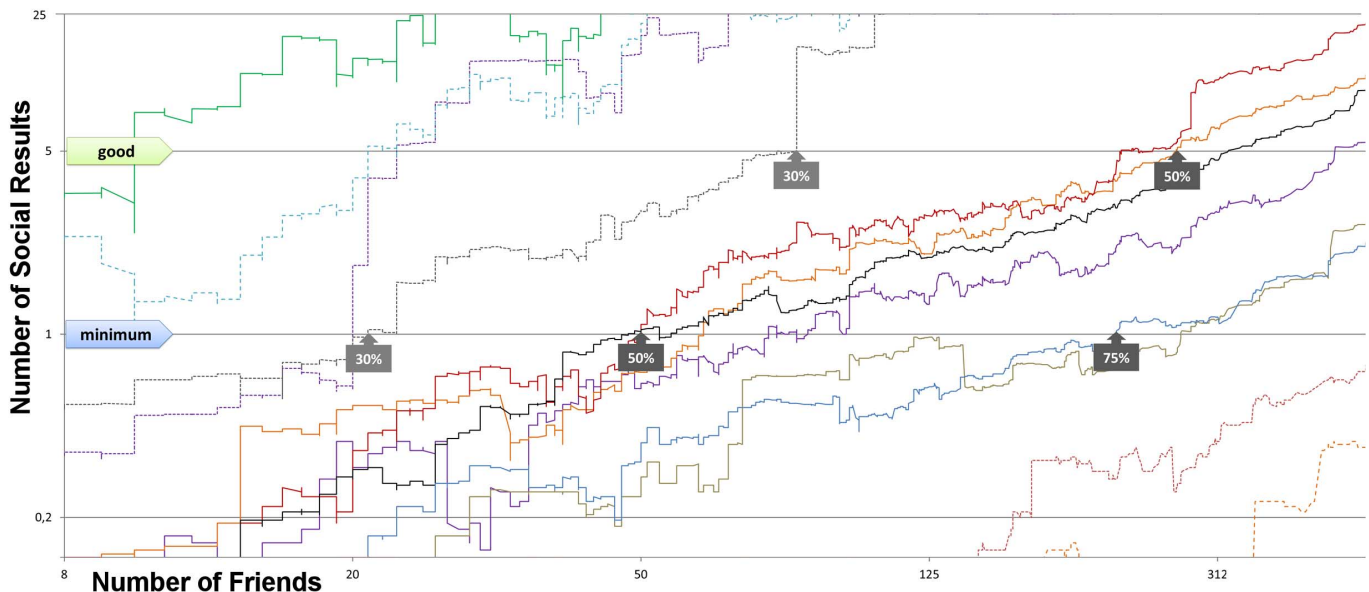
Users:	John243	Chris Rowlan
# of friends	79	59
Number of social results for the query:		
„Winter“	67	133
„SEO“	21	162
„Pizza“	9	13
„Buy Car“	2	7
„Android App“	3	0

**Result 2:**

- The number of link recommendations available for a user depends on his choice of friends as well as their activity.

**6.2 Social Search Queries**

Hypothesis 3 states that there is a minimum number of friends the user needs for social search to produce an adequate number of results. To investigate this hypothesis, we have defined two

**Figure 5. Social Result Count for one-word queries**

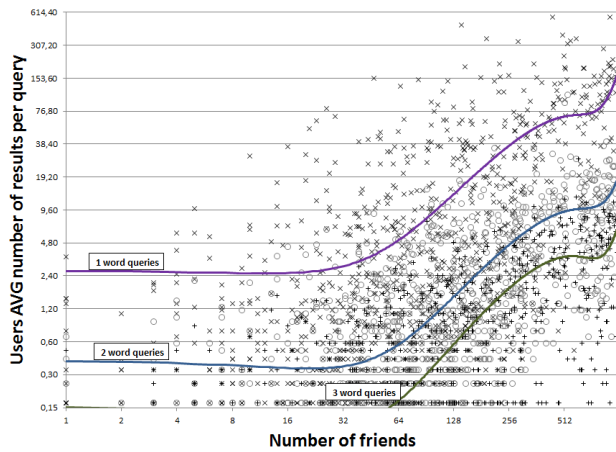


Figure 6. Results by Query Term Length

threshold values. An average of one social result per query will be the minimum. Below this value, social search does not work. An average of five social results (out of 10) in turn is good. This is because half of the results on the first page are recommended by friends.

From the search log, we have randomly selected 36 queries with a length of one to three words and have classified each one as informational, navigational, and transactional by hand. We then have executed these queries for the social graph of each of our 2,385 test users. Figure 5 represents each search query as a line. Our data shows that a user with 20 friends will touch the minimum (one result) for about 30% of all one-word queries. To check whether social search “works” on average queries, we define that at least 50% of the queries must return one social result or more. With this definition, a user needs about 50 friends for social search to work at a minimum level and about 300 friends to work well. That is, the longer and more complicated the queries become, the more friends someone needs to reach the minimum number of results.

#### Result 3:

- For every other query, the minimum threshold is reached with 50 friends and the optimum with 300 friends.
- For more complex queries, the thresholds are reached only in a quarter of all cases, even with 300 friends.
- The number of social search results strongly depends on the number and the degree of activity of friends.

Hypothesis 4 states that social results also depend on the length and type of the query. To evaluate this hypothesis, we have calculated the average result count for different query lengths. Figure 6 plots the average number of social results against the number of friends per user. We found that there tend to be fewer social results for longer queries.

Next, we look at the different query types. Figure 7 shows the average number of social results for different types. As expected, informational queries perform best and navigational ones worst. Surprisingly, transactional queries do not benefit from social search. In a second analysis, we observe that results of actual (logged) queries outperform the simulated queries executed for each user by a factor of almost two. This can be explained by the implicit knowledge of context of the user about topics in his environment: This means that the possibility for such queries to deliver results is higher than on random queries.

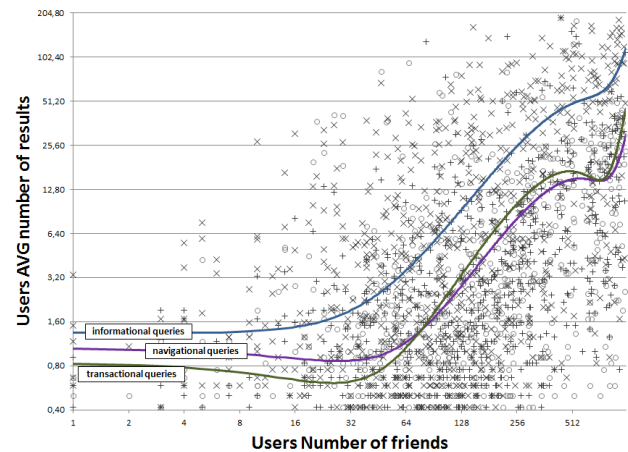


Figure 7. Results by Query Type

#### Result 4:

- Social search tends to work better on shorter queries.
- Social search works best on informational queries, while navigational and transactional ones are more difficult.
- In social search, the performance of real queries is significantly higher than that of random ones because the user already knows which result he can expect.

### 6.3 Search Behavior

Our final question is whether result ranking based on the Social Relevance Score (SRS) increases the search quality, compared to classic full text search (Hypothesis 5). To study this question, we have logged the search sessions of our test users (with their permission). We record how many results the user has clicked on, which position the selected result was at, and whether or not it has been a friend recommendation.

Table 3. Analysis of SRS Ranking Performance

Results (random sample)	No. Search Sessions	Ranking Position	Result Click Count	Search Time
Clicks on Social Results	117	2.92 (± 6.63)	1.14 (± 0.49)	13.56 sec (± 19,90)
Clicks on Full text Results	56	5.70 (± 8.26)	1.39 (± 8.26)	24.64 sec (± 84,14)
Difference	-	<b>+2.78</b>	<b>+0.25</b>	<b>-11.08 sec</b>
Confidence	-	<b>0.14 %</b>	<b>0.55 %</b>	<b>6.19 %</b>
Level(T-test)		(t = 2,93)	(t = 0,32)	(t = 1,39)

For this analysis, we have examined a random sample of 173 of all search sessions recorded where users actually clicked results. Table 3 shows that, in these search sessions, most users decided to click social results. Further, users have deemed social results more relevant. Over all sessions, the *RankingPosition* of clicked results was 2.92 for social and 5.7 for full text searches. We have verified the statistical relevance of this data with a T-test.

The same holds for the number of clicks a searcher needs to reach a satisfying result. While the *ResultClickCount* is 1.39 for full text



results, it is only 1.14 for social results. A T-test confirms that this difference is significant. While we could also observe a decrease in the time a user needs to decide which result to click, this result has not been significant.

Another approach to measure which results users prefer is to look at the alpha slider. This slider allowed users to adjust the search results to be more social or more objective. Its default value is 0.5. Most of the users (77.3%) have used the default setting. Of those who have adjusted their search, 60.79% have chosen to shift the slider towards “more social” results. The average value of the manually adjusted sliders has been 0.64 (towards social) with a standard deviation of 0.11. One could argue that the users are just curious about the new feature. This is not the case. We could verify this with a control group of students. We explicitly advised them to test different alpha settings and to only click on a link in the best result set. Therefore, we conclude that users indeed tend to find social results more compelling.

#### Result 5:

- When combining full text results with social relevance by SRS, users tend to find interesting results earlier (2-3 positions).
- Users are more satisfied with social search results: They need 0.25 clicks less to find a suitable item.
- When confronted with the option to adjust their search to be more social or more objective, 61% choose the first.

## 7. CONCLUSIONS & FUTURE WORK

### 7.1 Conclusions

In this paper, we have studied the question whether leveraging data from social networks, status messages in our case, improves search quality. We have analyzed existing approaches to social search and have identified the patterns of “user engagement” and “social trust”. We combine these patterns into a ranking metric called *Social Relevance Score* (SRS). To evaluate this metric, we have implemented a working search engine [www.social-search.com](http://www.social-search.com) on top of the social bookmarking platform [folkd.com](http://folkd.com).

During the course of our field study, over 2,300 people connected their [folkd.com](http://folkd.com) accounts with Twitter or Facebook. These users granted access to their social graphs and allowed us to crawl over 24 million links from their network of over 480,000 friends. We found that, on average, each friend creates one link recommendation per day. One should have more than 50 friends for friend-filtered social search to deliver significant improvements. The results of our field study show that social search results are more relevant: They decrease the time required for the search process and increase the user satisfaction.

### 7.2 Future Work

Some of our testers have participated in a survey after finishing the field study. Most of them have stated that the perceived value of receiving personal recommendations by friends is much higher than just getting “anonymous” results. Promoting more of these intangible values is interesting to improve the system in terms of user satisfaction.

In our implementation of [social-search.com](http://social-search.com) we have put much emphasis on minimizing the number of steps between accessing the site for the first time and getting to an individual search result. To improve this even further we will work on processing external

user data more quickly. We are also thinking about a process to predict which recommendations a user might also like based upon sample data. By identification of key users, we might introduce information hubs, which can act as trusted entities for users who do not have many friends.

## 8. REFERENCES

- [1] Schenkel, R., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Parreira, J. X., and Weikum, G. 2008. Efficient top-k querying over social-tagging networks. *SIGIR '08*.
- [2] Horowitz, D. and Kamvar, S. D. 2010. The anatomy of a large-scale social search engine. *WWW '10*.
- [3] Lerman, K. and Ghosh, R. 2010. Information contagion: an empirical study of the spread of news on digg and Twitter social networks. *ICWSM '10*.
- [4] Yanbe, Y., Jatowt, A., Nakamura, S., and Tanaka, K. 2007. Can social bookmarking enhance search in the web? *JCDL '07*.
- [5] Page, L., Brin, S., Motwani, R., and Winograd, T. 1999. The PageRank citation ranking: Bringing order to the Web. *Stanford InfoLab*.
- [6] Evans, B. M. and Chi, E. H. 2008. Towards a Model of Understanding Social Search. *CSCW '08*.
- [7] Heymann, P., Koutrika, G., and Garcia-Molina, H. 2008. Can social bookmarking improve web search? *WSDM '08*.
- [8] Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., and Su, Z. 2007. Optimizing web search using social annotations. *WWW '07*.
- [9] Rowlands, T., Hawking, D., and Sankaranarayanan, R. 2010. New-web search with microblog annotations. *WWW '10*.
- [10] Clements, M., de Vries, A. P., and Reinders, M. J. T. 2008. Optimizing single term queries using a personalized markov random walk over the social graph. *ESAIR*.
- [11] Gou, L., Chen, H., Kim, J., Zhang, X., and Giles, C. L. 2010. SNDocRank: document ranking based on social networks. *WWW '10*.
- [12] Ma, H., King, I., and Lyu, M. R. 2009. Learning to recommend with social trust ensemble. *SIGIR '09*.
- [13] Hotho, A., Jaschke, R., Schmitz, C., and Stumme, G. 2006. Information Retrieval in Folksonomies: Search and Ranking. *ESWC '06*.
- [14] Mislove, A., Gummadi, K. P., and Druschel, P. 2006. Exploiting social networks for Internet search. *HotNets-V*.
- [15] Cleverdon, C., Mills, J., and Keen, M. 1966. Factors Determining the performance of indexing Systems. Aslib Cranfield research project, Vol. 2.
- [16] Jansen, B. J., Booth, D. L., and Spink, A. 2008. Determining the informational, navigational, and transactional intent of Web queries. *Information Processing and Management*, 44.
- [17] Robertson, S. and Sparck Jones, K. 1976. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27.
- [18] Bastian, M., Heymann, S., and Jacomy, M. 2009. Gephi: an open source software for exploring and manipulating networks. *ICWSM '09*.
- [19] Truls A. Bjorklund, Götz M., and Gehrke J. 2010. Search in social networks with access control. *KEYS '10*.