

2011

Trust-Based Techniques for Collective Intelligence in Social Search Systems.

Pierpaola Dondio
Trinity College Dublin, Ireland

Luca Longo
Technological University Dublin, luca.longo@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/scschcombk>



Part of the [Computational Engineering Commons](#)

Recommended Citation

Donio, P. & Longo, L. (2011). Trust-based techniques for collective intelligence in social search systems. *Next Generation Data Technologies for Collective Computational Intelligence*, p.113-135. doi:10.1007/978-3-642-20344-2_5

This Book Chapter is brought to you for free and open access by the School of Computer Sciences at ARROW@TU Dublin. It has been accepted for inclusion in Books/Book Chapters by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)

Chapter 5

Trust-Based Techniques for Collective Intelligence in Social Search Systems

Pierpaolo Dondio and Luca Longo

Abstract. A key-issue for the effectiveness of collaborative decision support systems is the problem of the trustworthiness of the entities involved in the process. Trust has been always used by humans as a form of collective intelligence to support effective decision making process. Computational trust models are becoming now a popular technique across many applications such as cloud computing, p2p networks, wikis, e-commerce sites, social network. The chapter provides an overview of the current landscape of computational models of trust and reputation, and it presents an experimental study case in the domain of social search, where we show how trust techniques can be applied to enhance the quality of social search engine predictions.

1 Introduction

A key issue to the success of collaborative decision support systems, and indeed to any effective analysis of collaboratively generated content, is the reliability and trustworthiness of the entities involved. As user-generated content is no more regarded as a second-class source of information, but rather a complex mine of valuable insights, it is critical to develop techniques to effectively filter and discern good and reliable content. *The Wisdom of the Crowd* is not always sufficient to support good decisions, and many situations require the ability to spot the *Wisdom in the crowd*. One of the main challenges concerns how to effectively mine a large set of complex data affected by a great level of noise, represented by non-pertinent, untrustworthy or even malicious data. The proposed solution has to resist malicious attacks, spot low quality information and preserve privacy. Computational model

Pierpaolo Dondio

Department of Computer Science and Statistics, Trinity College Dublin

e-mail: pdondio@cs.tcd.ie

Luca Longo

Department of Computer Science and Statistics, Trinity College Dublin

e-mail: longol@cs.tcd.ie

of Trust and Reputation appear to be essential candidates to enhance and support an effective analysis of web activity. These mechanisms could help filter, interpret and rank web-users' behaviour to assign the relevance of web-search results and deliver the most reliable and adequate content. Similarly, they may be helpful in defining user-based anti-spam techniques, in supporting web-analytics applications that mine only trustworthy sites and users' activity, and helping users' segmentation and decisions support tools for online marketing. This chapter presents the current landscape of computational trust models, and describes how such techniques can be used to enhance the quality of collective computed intelligence.

Computational models of the human notion of trust have emerged in the last decade with the aim of predict and quantify the trustworthiness of digital entities in open and collaborative environments. The word Trust is used here to define a quantifiable prediction about user's expected ability to fulfill a task. When applied to computational intelligence, a trust computation helps predicting which peers will likely produce useful and reliable content for the community. A level of trust in our context is therefore a concept that overlaps competence, expertise and reliability. In particular, we present an experimental study case where we apply a trust function in a collaborative distributed domain. The domain chosen is Social Search, a fast-growing information retrieval paradigm where documents are ranked according to how the web-community is sharing and consuming them. Social search represents an ideal study case due to its collaborative, decentralized and large-scale nature. Our experimental study shows how trust techniques improve the quality of Social Search engines, confirming their central role in deploying effective collective intelligence in the age of Global Computing.

This chapter is organized as follows: Section 2 introduces the core concept of collective intelligence and distributed decision making, section 3 describes the current landscape of computational models of trust, and how trust models can be used as decision support tools. Section 4 introduces the concept of Social search, describing briefly the main trends and challenges of this paradigm. Section 5 describes a practical social search technology used in our study case along with the definition and implementation of a trust model for social search. Section 6 describes our experimental results and section 7 concludes the chapter underlining future directions.

2 State-of-the-Art: Distributed Decision-Making and Collaboration

Collaboration is a process where people interact with each other towards a common goal, by sharing their knowledge, learning and building consensus. This concept does not require a leadership figure and it can deliver good results if applied in decentralised distributed systems. The Internet is the most popular scenario where entities are widely distributed, without any kind of authority. The Web 2.0 is the evolution of the World Wide Web. This term refers to applications in which users can contribute independently, sharing information towards new collaborative architectures, creating worldwide network effects. The contribution is intended as a process

where an entity, usually an individual, provides a judgement about another entity, either digital or human, by using specific graded relevance systems such as numbers, letters, descriptions. Wikipedia is a good example in which a good degree of collaboration can be achieved. Here humans collaborate towards the development of an open encyclopaedia on a distributed world wide scale, by creating and editing articles about a disparate range of fields. The fact that this online encyclopaedia is written by an open community of users around the world and the majority of its articles can be edited by anyone with access to the Internet underlines the intrinsic degree of collaboration; several studies suggest that its content can be as accurate as other encyclopaedias [42].

The collaboration applied to Web 2.0 applications supports a new kind of shared intelligence, named *Collective Intelligence*. Here users are able to generate their own content building up an infrastructure where contributions are not merely quantitative but also qualitative [43]. *Collective Intelligence* has been defined in several ways. A shared agreement suggests that it is a group/shared intelligence that emerges from the collaboration and competition of many entities, either human or digital. Collecting judgement from a large group of people allows drawing statistical conclusions about the group that no individual would have known by themselves. The resulting information structures can be seen as reflecting the collective knowledge of a community of users and can be used for different purposes. For instance, as in collaborative tagging systems such as Del.icio.us¹, where users assign tags to resources and Web-entities shared with other users, the emerged community's knowledge, due to users' interaction, can be used to construct folksonomy graphs, which can be efficiently partitioned to obtain a form of community or shared vocabulary [38].

Although techniques for Collective Intelligence existed before the advent of the Internet, the ability to capture and collect information from thousands or millions of people on the World Wide Web has accelerated the proposal of new practical technologies aimed to provide applicable intelligence in the decision-making process. *Social Search* may be considered one of these technologies, an application of *Collective Intelligence*. Here multiple entities' behaviour is taken into account in order to deliver a usable supporting tool for classifying and ranking web-resources, therefore predicting web-users' requirements.

3 Computational Trust

Trust and Reputation are two indisputably recognised relevant factors in human societies. Several studies have been carried out in several fields: psychology [20], sociology [4], economy [6] and philosophy [16]. Computational models of trust emerged in the last decade with the aim of exploiting the human notion of trust in open and decentralized environments. According to Luhmann [29], trust is adopted by humans to decrease the complexity of the society we are living by using delegation. Trust has emerged as a key element of decision-support solution helping agents in the selection of good and trustworthy collaborative partners, in the identification

¹ <http://www.delicious.com>

of reliable pieces of information or as part of soft-security applications. Several definitions of Trust have been proposed. As suggested by Gambetta, trust is a prediction (subjective probability) that the trustee entity will fulfil trustier' s expectations in the context of a specific task [13]. A typical computational trust solution follows the high-level architecture depicted in 1, modelled after the Secure trust engine [9]. In a typical distributed environment, an agent - the trustier - is acting in a domain where he needs to trust other agents or objects, whose ability and reliability are unknown. The trustier agent queries the trust system to gather more knowledge about the trustee agent and better ground its decision

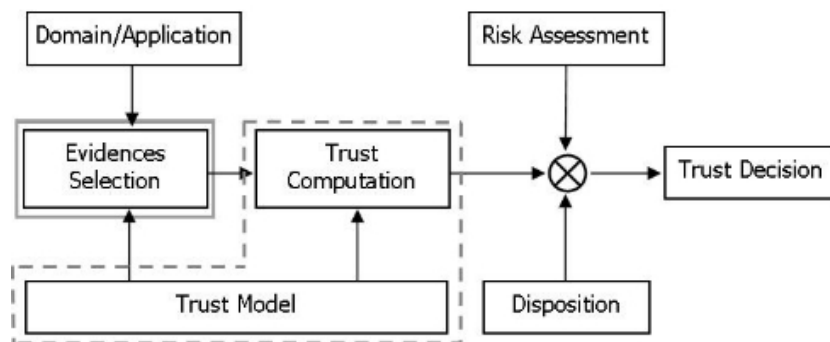


Fig. 1 A computational trust solution.

A trust-based decision in a specific domain is a multi-stage process. The first step is the identification and selection of the appropriate input data. These data are in general domain-specific and identified through an analysis conducted over the application. We refer to this process as evidence selection and to the inputs used to compute trust as trust evidence. Evidence selection is driven by an underlying trust model that contains the notion of trust on which the entire system is centered. A trust model represents the intelligence used to justify which elements are selected as trust evidence, why some elements are selected and other discarded, and it informs the computation over the selected evidence. A trust model contains the definition of the notion of trust, its dynamics, how it evolves over time and with new evidences, and the mechanisms of trust used in the computation. After evidence selection, a trust computation is performed over evidence to produce trust values, the estimation of the trustworthiness of entities in a particular domain. A trust computation requires the formalization of a computable version of those mechanisms defined in the trust model. Examples of such mechanisms are the past-outcomes one, reputation and recommendation, but also temporal and social factors, similarity, categorization and so forth. For instance, a classical trust system uses two set of evidence: recommendations and past experience. Each of them is quantified separately and then aggregated into a final value. In this final aggregation stage, exogenous factors such as risk

and trustier's disposition can also be considered. The output is presented as quantitative trust values and as a set of justifications. Fig 1 depicts the main component of the trust system described so far.

3.1 *Computational Models of Trust*

Current trust systems can be divided in the following macro-areas:

- Security-oriented approach
- Explicit-feedback systems
- Rule-based systems
- Probability-based systems, or past-outcomes, implicit learning systems
- Game Theoretical
- Cognitive models and computational trust models

With security oriented approach we intend a situation in which the focus is still the on the possession of a valid object, usually a credential, that allows an entity to access some resources and therefore to be trusted. Questionably, they are not trust system but security systems. Examples are PKI infrastructure, with third trusted party of decentralized as in a PGP scenario. A dedicated infrastructure, separated from the application, is in place to gather the required object and transfer among the peers community. The trust intelligence encoded in such systems is limited to the transitivity of trust, that means the fact that trust is propagated through a chain of trusted individuals. Transitivity is the mechanisms at the core of social networking applications [15], with the difference that what is propagated is social information, usually a level of acquaintance between two entities. Information sharing is also at the core of feedback systems, such as reputation or recommendation systems. In such systems users share recommendations in order to have a better idea of their peers. While reputation is a visible global value, expressing the consensus of a group [40], recommendation is an opinion privately shared. Advanced recommendation systems consider the level of trustworthiness of the recommender's peers, or better their ability to provide recommendations; these systems consider situational factors [18], the noise associated to the length of the chain [40], the consensus or conflicts among various sources [18]. In a rule-based system, trust is a collection of rules identified by domain experts that deliver the trust solutions. In the past-outcome paradigm, or direct experience, trust is computed using evidence that the trustier gathered directly from previous interactions in order to predict trustee's future behaviours. A clear definition of this computation, and the correlated notion of trust, is the one produced by the research group Trustcomp.org: 'Trust is a prediction of future behaviour based on past evidences'². There are many different incarnations of the past outcome paradigm, but they all share a common basic scheme. The central notion is that a trust value is computed using the outcomes of all the pertinent past interactions. The value is updated when a new interaction occurs, proportionally to the outcome of this interaction. Examples are found in Quercia's model [37],

² Trustcomp online community, www.trustcomp.org

Wang p2p trust engine, the Secure project. In the Probability-based approach trust is represented and manipulated (predicted) as a probability distribution function that typically models the expected behaviour of a trustee. Advantages are a clear (but limited) semantic meaning and effective computational tools. In the use of beta-distribution and the Bayesian Inference, probability offers one of the most powerful tools for computing trust, where probability becomes not only a meaningful trust representation; it goes further, offering also mechanisms for updating and learning trust. The beta-distribution is a family of pdfs used to represent a distribution over binary outcomes. A beta-distribution is completely defined by two positive numbers. The two parameters define completely the expected value and the shape of the distribution. As an example, figure 2 presents a beta distribution with the value of $(1, 1)$ on the left and $(8, 2)$ on the right.

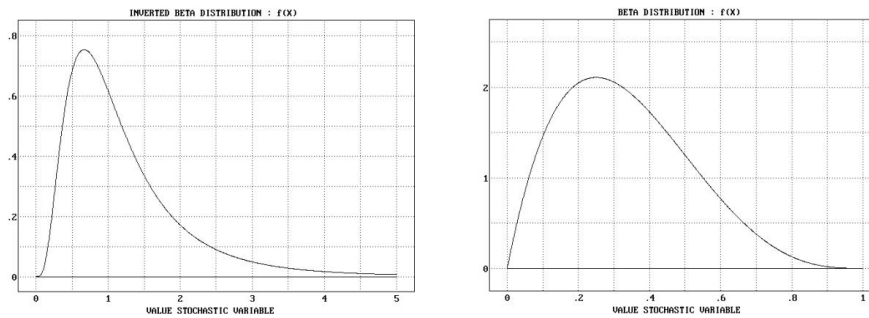


Fig. 2 Beta Distributions

This behaviour maps a representation of a trust value based on evidence. Usually, the two parameters a and b are the numbers of positive and negative evidence regarding the trustee, and the pdf distribution characteristics (expected value, variance) are used for trust values computation and uncertainty assessment. The method is used by Josang [18] or in [40], where r and s are respectively the good and bad evidence regarding a trustee, and $a = r + 1$ and $b = s + 1$ define the corresponding beta pdf. Referring to figure 1, when no information is available about an agent, ($r = s = 0$), the beta distribution $(1, 1)$ is uniform: no value is more likely than others and the uncertainty is at its maximum value. When, for instance, an agent holds 7 positive pieces of evidence and 1 negative, the corresponding beta distribution $(8, 2)$ is distributed around the average value of 0.8 with a small variance. In the Game Theoretical approach, as described by Sierra and Sabater in [40], trust and reputation are the result of a pragmatic game with utility functions. This approach starts from the hypothesis that agents are rational entities that chose according to the utility attached to each action considering others' possible moves. Action could be predicted by recognizing an equilibrium to which all the agents are supposed to tend in order to maximize their collective utility. The Game Theoretical approach in trust can also be encoded in the design of the application. In this case, the application is

designed so that trust is encoded in the equilibrium of the repeated game the agents are playing. Thus, for rational players trustworthy behavior is enforced.

Finally, a formal notion of trust has been formalized in many rich trust models, notably cognitive models of trust. These models present an articulated and composite notion of trust, and their aim is to define trust as a computational concept. Marsh's first model of trust is still a benchmark. His work gives many insights on the notion of trust as it emerges from social science and it provides a formalization of key aspects, such as basic trust, disposition, reciprocity, situational trust, the concept of a cooperation threshold to start an interaction. A cognitive model of trust defines the mental processes, mechanisms and dynamics of trust. These models stress the nature of trust as a complex structure of beliefs and goals, implying that the trustier must have a "theory of the mind" of the trustee [8]. Trust becomes a function of the degree of these beliefs. Cognitive models present a rich notion of trust, and reject the reduction of trust to a probability-based computation, that is seen as a simple and limited approach, as described by Castelfranchi and Falcone in [8]. Dondio [11] proposes a model of trust/reputation based on defeasible reasoning and knowledge engineering. This model considers the action of evaluating entity's trustworthiness an argumentation process. The form of such argumentation is represented by a defeasible reasoning semantic. A knowledge-based model of trust, as it emerges from social science, provides the content of each argument involved in the trust computation. The model is applicable to a large series of Web 2.0 applications such as Wikis and Online Communities.

4 Social Search

The phenomenon of *Social Search* has been acquiring importance in the World Wide Web with the proliferation of large-scale collaborative digital environments. A social search engine is a type of web search technique that takes into account the interactions or contributions of end-users in order to enhance the relevance of web-search results. The main advantage of such a system is that the value of Web-pages is determined by considering the end-user's perspective, rather than merely the perspective of page authors. This approach takes many forms, from the simplest based on sharing bookmarks [14], to more sophisticated techniques that combine human intelligence with computational paradigms [7]. The recent *Social Search* approach contrasts with established algorithmic or machine-based approaches such as the one of the leading searching engine, Google, whose Page-Rank algorithm [34] relies on the link structure of the Web to find the most authoritative pages. A key challenge in designing *Social Search* systems is to automatically identify human values in the Web. In other words, instead of analysing web-links among web-pages, social search aims to analyse human behaviour. As a consequence, capturing and collecting humans' values is the first step towards the inference of the relevance of web-resources. As mentioned before, a Social Search engine ranks web-resources according to how users of a community consume and judge those resources in relation to some searching needs. A particular practical problem for any potential

solution based on gathering end-users' behaviour on a web-page is that they tend to be resistant to explicit and invasive techniques and as a consequence it is not easy to generate strong recommendations. In contrast, implicit feedback techniques capture activity performed by users over Web-pages indirectly.

As suggested in [25], there are two ways for providing judgements:

- explicitly: users can provide feedback using a specific metric, by using letters, numbers or complex structures. The most popular examples are eBay³ and Amazon⁴ where buyers and sellers can rate transactions using a given graded system;
- implicitly: implicit judgements are automatically inferred by analysing users' behaviour while performing a specific task. Their behaviour is captured by data-mining software that generates logs, raw data that need to be analysed, filtered and aggregated in order to extract meaningful information. A web-proxy monitor is an example of logger: it is a piece of software embedded in a web-proxy server, a special computer that acts as an intermediary for requests from others computers seeking web-resources. This software can capture web-site requests, URLs, request time, IP addresses, all potential behavioural information. A lower-level logger is represented by browser-plugins or add-ons, special software able to capture events such as scrolling, reading time, bookmarking, cut-paste, form filling, saving pictures, generated by Web-browsers such as Internet Explorer or Mozilla Firefox. These browser-events are all considered relevant implicit sources of user preferences [21].

Independently from the solutions adopted, whether explicit or implicit, there is a key problem to take into consideration: the trustworthiness of those entities who provide judgments. If entities who provide recommendations are malicious or untrustworthy, the resulting quality of the rank of web-resources is negatively affected. Computational trust techniques can be successfully applied in the context of search to enhance the quality of social Search engines. Here a trust module may be integrated in order to filter data and to make an engine's predictions more accurate. The users' level of trust, for example, may be assessed by considering their expertise in gathering information within the Web, and their ability to fulfil a searching problem. In other words, trustworthy users are the ones able to find the most relevant information when they need it.

5 Computational Trust to Enhance Social Search Ranking: A Practical Study-Case

We have developed a prototype of a search engine based on user-activity containing dedicated algorithms to rank pages, identify search sessions, query boundaries and group similar queries. The Prototype can incorporate a Trust computation to rank each peer based on its activity and use this value to weight its contribution, giving more importance to the most trustworthy peers. The Prototype components are:

³ <http://www.ebay.com>

⁴ <http://www.amazon.com>

1. **Prototype Plugin:** a software component responsible for monitoring a user's activity, storing it locally in a structured file. The plugin captures all the major browser events and generates a well-structured XML string, easy to parse for different purposes. It contains the activity occurred in each window and each tab of the browser; it saves the URL and the title of the opened web-pages along with the start time, the finishing time and the focused time. It gathers the main events that may occur during an Internet session, with the related time-stamp such as bookmark, printing, submitting a form, saving as, cutting, pasting and so forth. The logger also triggers an event every n-seconds of inactivity (set to 10 seconds). Furthermore, the logger traces users' searching sessions. Each time a user submits a query to the Google search engine the logger stores the keywords used, and the ordered list of the search engine results for the query, along with the pages browsed in that search session, and these are identified by analysing the outgoing links from the search engine result page.
2. **Prototype Engine:** a software component, installed locally with the Prototype Plugin or remotely connected to the plugin, that is responsible for collecting the data generated by the plugin and processing them. The engine is composed by three procedures:
 - **Session/Queries Identifier:** this algorithm finds the boundaries of a search session, from the starting query to the last page of the last query of the session, the set of pages relating to each query of the session, it interconnects queries belonging to the same session.
 - **Evidence Selection:** this software component is responsible for interpreting the raw user data coming from the plugin and identifying activity patterns that will be used in the following rank computation.
 - **Local Evidence-Based Computation:** this algorithm processes locally the information extracted from the evidence selection components and it generates indicators regarding the pertinence of each page to the query it belongs to.
3. **Reasoning Engine:** this component is responsible for computing a rank for each page and each peer, and connects queries and search session together. It is composed of:
 - **Trust Computation (peers):** this algorithm processes users' activity and assigns to each peer a trust value that measures the peer's ability to perform and complete a search session.
 - **Rank Computation:** this algorithm performs the computation of the global indexRanking ranking for each page browsed in the context of the specific query and search session by processing the structured evidence (the arguments) identified by the Prototype Engine. The component takes the structured user data of each peer as an input and computes a global ranking for each page in the context of a specific query and session.
 - **Model Definition:** where users can edit their own models.

4. Prototype DB: this database contains information about rank pages, queries, search session; peers trust value in a structured way. Data are organized by queries interconnected to each other and by peers. It is composed of:
 - Query Clustering: this component aggregates, links and clusters queries and groups of documents.
5. Prototype Interface: the end-user interface is used by a peer to query the Prototype DB via the Results Manager component.

5.1 The Functioning

A peer connects to the Prototype Community. The Prototype Plugin continuously monitors users' browsing activity, saving it into a local Raw Activity Data (RAD). Every time a peer submits a query to a search engine, the Prototype Plugin saves information related to the set of keywords used and the list of documents proposed by the search engine as result of a search query. The local raw activity file is periodically analyzed by the Prototype Engine that extracts information about a peer's activity for each document. The information extracted is organized into a complex set of evidence that forms an argument against or in favour of the pertinence of the page browsed in the context of a search session. The Prototype Engine sends the processed data, the Structured Activity Data (SAD) to the Reasoning Engine. The Aggregator computes a rank for each page in the SAD by means of a reasoning process. Data about the global activity of the Prototype community is retrieved by the reasoning engine to perform its computation. Each ranked page is saved in the Prototype Database, following a query-based data organization. The information about page ranks, along with the arguments used, is saved into the Prototype DB. Data are organized by query or keywords. A cluster matching component periodically organizes the information contained in the Prototype DB by clustering and grouping queries. The component adds logical links to queries that are considered similar or relevant using a page- and activity-based clustering approach. When a peer starts a search session and he wants to exploit the Prototype Page Rank, the query is sent to the Prototype Database which retrieves the pertinent documents for the query.

5.2 Computational Trust to Enhance the Social Search Engine

In this section, we present an experimental case study that shows how computational trust can be used as a form of collective intelligence to enhance social search, chosen as an example of distributed collaborative application. The ability to search an increasingly large and noisy Web has become a non trivial expertise, involving cognitive and practical abilities, and a familiarity with the browser technology. It is reasonable to assume that web users will exhibit a different level of expertise, directly linked to their ability to correctly identify the most pertinent information.

Therefore, there are reliable and trustworthy users - providing pertinent pages and meaningful results - and untrustworthy ones - generating incorrect results or even noise. Therefore, in our context, trustworthy users capable of finding the most relevant information, that means expert search users. Three main factors are involved in the ability of users to deliver good search results:

1. Cognitive skills, that means, the ability to read quickly, scan information, analytic/global thinking abilities;
2. Search Experience, that means, the familiarity with searching and browsing technology;
3. Domain specific knowledge, that means the expertise and interests that users might have in specific topics.

In this experimental study we define a trust function (or expertise function) to indicate reliable searchers modelled around the second factor. The other two factors, complementary to the second factor, are well-studied and are not discussed or used in the definition of the trust function. Our discussion will be mainly descriptive, focusing on the main concept rather than the technical details. Another important limitation to mention is the fact that this study only considers navigational queries (queries pointing to a specific piece of information) and not to informational query (open-ended). The following picture shows the trust computation is integrated into the Prototype Social Search Engine.

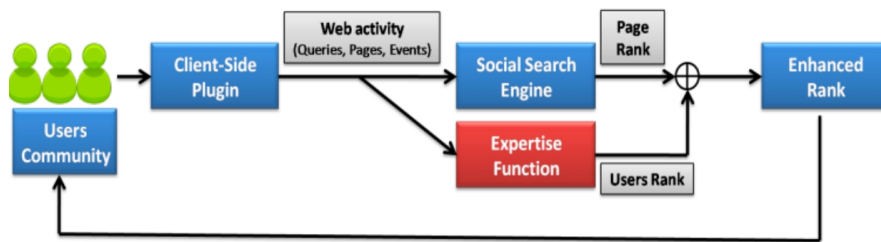


Fig. 3 Search Engine and the expertise Function

Our Prototype is used to gather data (client side plugin). We study how to enhance these search engines with an expertise function that assigns a level of searching expertise to each peer, and use this level to make engines' prediction more accurate. In order to keep more generality, we do not study the effect of the introduction of the trust function directly on our results, but rather we test the effect over a series of performance criteria commonly used by the majority of social search engine. Our trust function computes a level of expertise in the domain of searching based on the three concepts of browsing experience, competence and quality of past-performance. The hypothesis is that, by relying only on the super-users identified, the ability of a social search engine to spot relevant pieces of information in relation to a query will be

enhanced. The enhancement is expected for two reasons: first, engines will generate a rank based on a smaller set of data; second, the rank will be based on a subset of information of better quality. We present a case-study performed over a set of 90 users, using their search logs collected for a period of 3 months and the results of lab-based search problems.

5.3 *Leveraging Users' Features to Rank Pages*

As discussed earlier, there are three main user features that certainly impact page ranking, namely:

1. Domain Expertise
2. Search Experience
3. Cognitive Style and Ability

Our trust function focuses on the second factor, although the other factors could also be embedded into a trust function, since they are complementary features needed for a complete exploitation of users' features. The first feature, Domain Expertise, considers the user's familiarity with the subject on which he or she is searching. The third feature, Cognitive Style, concerns the cognitive ability of the searcher and the type of intelligence. Searching effectiveness is influenced by the way users assimilate new information, users ability to read, memorize, scan documents, analyze text, images, colours. The Search Experience of users - the core of this chapter - can be leveraged to improve search engines ranking. Users who have a better understanding of the breadth of a search engine's capabilities have more ways to go about finding information. Knowing about Boolean operators, exact strings, filtering controls, and having proven strategies to exploit search, allows you a much richer toolset at your disposal. No quantitative study has been proposed to study the impact of search experience on web ranking. In [3] the authors focused on tasks such as classifying the users with regard to computer usage proficiency or making a detailed assessment of how long it took users to fill in fields of a form. The work provides hits and evidence for our analysis, since it was the IT familiarity with the browsing technology to be analyzed as we investigate. The information is yet not used to improve the ranking but rather to investigate the universe of web users and classifying patterns. According to a study performed by Hastings Research in the SEO world⁵ the use of Boolean operators and other advanced query filters and connectors proved to narrow search results and improve the relevance of results. Some data samples suggest that these search skills are known to a relatively small percentage of users, but their effectiveness has been proved⁶. Anyway, the works clearly show that relevance is improved, and therefore the use of such advanced features could be used as a proxy to identify expert reliable users. The work performed by Augrihm [2] is described in the next section.

⁵ Hastings Research databases of real-time searches and web server logs, 1995-2008.

⁶ Hastings Research databases of real-time searches and web server logs, 1995-2008.

5.4 Performance Criteria

In this section we provide a set of criteria to test the impact of our expertise function on search ranking. The way implicit feedback algorithm processes user information suggests that ranking is obviously improved the more users find relevant information, and indeed by the way relevant information is found. Search engine works better when relevant information is found minimizing noise, represented by useless extra clicks, page browsed, queries reformulated and so forth. Even when a query is successful, situations in which the noise is minimized should be rewarded. The criteria we used to assess the quality of the information provided for a specific search problem are:

1. Percentage of peers that successfully completed the (navigational) query, since the more users reached the web page containing the information searched, the more chances the page had to be ranked high in the social engine rank;
2. Number of pages visited for successful queries. An ideal query is the one where only the page containing the answer is visited, since there is no ambiguity in ranking that page high in relation to the query. On the contrary, the more pages are visited the more noise is introduced into ranking algorithms, leading to a higher probability that the correct page will be mixed with useless ones;
3. Number of queries submitted. Analogous to the previous criteria, a limited number of queries refinements helps to better identify the search problem the user is looking for, that means the search context, important information to feed query classification and similarity/matching algorithms, reducing the probability of ambiguity in the search problem definition.
4. Time spent for a query. For navigational queries, short time on a query is a plausible evidence that the information has been found. An ideal navigational query is the one where non pertinent pages are discarded quickly. Moreover, in navigational query type, it is common that a relevant page is quickly analyzed and registers low activity if the information is clearly shown on the page.

The trust function

The trust function used to identify trustworthy users encompasses 3 factors: past performance, experience and competence

Past performance factor

Past-performance can be implemented by looking at the outcome of previous search sessions and computing a level of expertise for each individual proportional to the number of search session completed positively. The factor is easily implemented for navigational queries type. The query is successful if the user can actually find that information, probably duplicated in more than one web-site. In order to compute a level of past-performance for each of our users, we defined a set of search problems S_n for which the piece of information I_n that satisfies the problem is known a priori (Dublin solves the query 'capital of Ireland'). Users are asked to perform S_n and

explicitly write the answer. We could presume that a user is asked to perform some test queries the first time he joins the social search network, following a similar pattern to other reputation-based social networks, or that a human-assisted procedure will analyze some of his queries. A user gets a full score, say 1, for each query if he gets the right answer, otherwise the proximity to the web page containing I_n is considered. If the user actually browsed the page containing I_n , without identifying I_n , the query failed but its browsing activity contains at least the correct page and a non-null score < 1 is assigned to the user. The user at least had provided the relevant page. With the same reasoning if the user browsed a page P one-hop close to the page containing I_n - assuming that P is not a search engine result page, usually it is a page in the same domain - a smaller not null value is assigned to the user. Past performance were computed using a beta distribution $\beta(a+1, b+1)$ where a is the index of success while b is the index of failure. a and b after the query $t+1$ are computed from the values of a and b based on the first t queries:

$$a_{t+1} = \begin{cases} a_t + 1 & \text{if the query was satisfied} \\ a_t + 0.5 & \text{if the user visited a page 1-hop from the page with the required information} \\ a_t & \text{elsewhere} \end{cases}$$

$$b_{t+1} = \begin{cases} b_t + 1 & \text{iff } a_{t+1} = a_t \text{ (query failed)} \end{cases}$$

A value of past performance TPP for a specific user U is given by the average value of the beta distribution $\beta(a+1, b+1)$ related to user U .

Enhancing past performance

The past-performance mechanism can be enhanced by considering not only the outcome of the query, but also the level of difficulty of the query. To consider the difficulty of a query we could consider:

1. the percentage of success of that query P_{suc} ,
2. time spent for a query T_q
3. number of pages visited PG_{tot} ,
4. number of queries reformulation N_{qr} for the query.
5. hop-distance of the solution to the search engine page N_{hop}^u

Note the overlapping with our evaluation criteria: the most difficult queries are those more difficult to be properly ranked, where the information searched is hidden and/or far from search engine results. The hop distance for a query is computed as the average distance of hops for the page containing I_n using the logs of all the successful queries. The level of difficulty for a query is computed by ranking all the five above indicators and by aggregating them, giving more weight to the P_{suc} as it is the only explicit indicator of query difficulty.

$$Qdiff = f(P_{suc}, T_q, PG_{tot}, N_{qr}, N_{hop}^u)$$

Experience factor

Our concept of experience is based on how much and for how long the user searched on the web. Experience has to do with the previous quantity of searches, rather than the quality of them, as it was for past-performance. Aligned to other studies, we presume that an experienced internet user has more chance to perform a correct query than an inexperienced one. Experience has been assessed in the following way. All the users were asked to install the plugin logger for a period of about 3 months. The data collected contains all users' browsing activity, including search sessions and queries executed. Using a special feature, we traced if a page belonged to a search session, i.e if there is a click-chain connecting the page and a search engine result page. Data are summarized in table 1, containing the average occurrences per user of all the browsing events monitored. A value of experience is computed considering the time each user spent on the web and the time spent searching. The time spent searching is approximated by the number of queries submitted and the total number of pages browsed in the context of a search session. Experience is therefore the aggregation of 3 ranked indicators:

- $Time_{web}$: total time spent on the web;
- N_q : total number of queries performed;
- P_s : number of pages linked to a search session.

$$Experience = f(N_q, Time_{web}, P_s)$$

Competence

Competence here is defined as the familiarity of users with search engine features and online search dynamics. While our concept of Experience is based purely on quantitative indicators, Competence analyzes the way and style a user performs in its web activity. Competence has not to be confused with the competence of a user in a particular topic. Here it is exclusively a metric of the competence of the user with the task of web searching. In order to compute competence we use the data collected in the 3-months monitoring of our user population (table 1). First, we select a set of events that can be plausibly linked to advanced searching skills, supported by the already cited studies [17]⁷. The events E_x were:

1. find in page
2. use of Boolean/special connectors,
3. use of exact string (1-3 all proven to increase searching relevance),
4. use of multitab/multiwindows browsing (evidence of ability to perform parallel analysis of information [12] [17])
5. percentage of results skipped next or/and above, evidence of ability of users to discard non pertinence information in advance [19].

The idea is now to assign a level of competence to each browsing event, and compute the user competence as the weighted average of the events he performed. The

⁷ Hastings Research databases of real-time searches and web server logs, 1995-2008.

competence weight of each event is the inverse of the probability that a user will perform that particular action, computed over the population's distribution of events (1). Therefore, less likely events have more impact on competence. The likelihood of a find in page (E1) was computed over the total number of pages browsed, the usage of Boolean connectors (E3) and exact string (E2) over the number of queries, the multi-tab and multi-window navigation (E4-14) over the total number of windows/tab opened. For instance, since 59 find in page have been recorded in 1254 pages, the probability that a user will perform that action is 0.047, and therefore the competence level of a find in page is 21.25. The formulas are shown below (E_{ix} = occurrences of events i , E_i^u occurrences of the event i for user u , N_p = number of pages, N_q = number of queries).

$$Competence_{E_1} = \left(\frac{E_{ix}}{N_p} \right)^{-1}$$

$$Competence_{E_{2,3}} = \left(\frac{E_{2,3}}{N_q} \right)^{-1}$$

$$Competence_{E_{4-14}} = \left(\frac{E_{4-14}}{N_p} \right)^{-1}$$

$$Competence_u(Competence_{E_i}, E_i)$$

Table 1 Browsing activity report

T	Browsing event	Avg. Occurrences per user
T	Total time (Hours)	2.14
N_p	Number of pages	4880
N_q	Query	539
P_s	pages belonging to a search query	1254
E_1	Find in page	59
E_2	Usage of Exact String	121
E_3	Logical Connectors or special command in the query	6.2
E_4	1 window (# of times user used only 1 window to browse)	4880
E_5	2 windows (# of times user used only 2 windows to browse)	291
E_6	3 windows (# of times user used only 3 windows to browse)	232.5
E_7	4 windows (# of times user used only 4 windows to browse)	57
E_8	5+ windows (# of times user used 5 or more windows to browse)	19
E_9	1 tab (# of times user used only 1 tab)	4880
E_{10}	2 tabs (# of times user used only 2 tabs)	1098
E_{11}	3 tabs (# of times user used only 3 tabs)	549.3
E_{12}	4 tabs (# of times user used only 4 tabs)	221.5
E_{13}	5 tabs (# of times user used only 5 tabs)	56
E_{14}	6+ tabs (# of times user used 6 or more tabs)	101

6 The Experimental Results

In this section we describe a first evaluation of the trust metric for social search. The controlled experiment was performed on a population of 93 users and focuses on navigational queries.

6.1 Navigational Queries

We defined a set of 20 navigational queries of different levels of difficulty that were executed by our population of users. For instance, the following are examples of queries of different levels of difficulty:

- Easy
 - Q1. Price of a ticket to enter Malahide Castle
 - Q5. Find if GB Shaw won a Nobel Prize in literature
- Medium
 - Q5. How much was the child benefit in Ireland in 2009?
 - Q9. How do you say 'when' in Latin?
- Hard
 - Q10. Number of 0-0 in the Premier League 2008-2009
 - Q15: When did the company Georgia Gulf double its price last summer in one day? On which news?

6.2 Enhancement of Past Performance

The past performance value is generated by the formula described in the previous section. Queries are ranked according to their degree of difficulty, quantified by Q_{diff} (section 5.4). Referring to the above set of 6 navigational queries, table 2 displays the value of Q_{diff} normalized from 0 to 1.

Table 2 Queries Difficulty Levels

Query ID	Q_{diff} normalised	P_{suc}
10	1	28%
15	0.95	44%
4	0.55	81%
9	0.5	88%
1	0.1	100%
5	0.05	100%

6.3 Baseline Results without Trust Function

We now compute our four performance criteria using all the population of 90 users. Table 3 shows the results for the top two most difficult queries (Q10 and Q15) and the others organized into 6 groups of 3 queries.

Table 3 Baseline results without trust

Query or Group	P_{suc}	N_p	N_q	T
10	28%	18.78	6.2	10'22"
15	44%	14.9	4.2	6'53"
G1	78%	13.77	3.4	6'05"
G2	78%	7.2	2.12	3'56"
G3	91%	4.56	2.6	2'23"
G4	91%	5.15	2.1	3'04"
G5	97.9%	4.89	1.78	2'03"
G6	100%	2.89	1.19	1'34"

6.4 Experiment 1 - with Past Performance

In this first experiment, we test the entire trust function, composed by enhanced past performance, competence and experience. The trust function used assigns more weight to the past-performance factor. User's U trust value SE_u is computed as:

$$SE_u = 2 \cdot T_{pp} + T_{comp} + T_{exp}$$

The 90 users were ranked according to their SE_u and normalized from 1 (the most trustworthy) to 0. In order to compute the past-performance value, the 20 queries were divided into train and test group of balanced difficulties, each containing 10 queries. The four performance criteria were now recomputed using a weighted average, using the trust value as the weight. Table 4 shows a comparison between the indicators for each query.

Results are encouraging and show the benefits of our expertise function in a social search context. The performance gain is evident for the most difficult queries. Q10 has now a success rate of 53%, from a baseline of 28%, 6 pages less visited on average, a slightly diminished number of query reformulation and a gain of 20% of time. This implies that a social search engine would have ranked pages in relation to that query better, with an increased likelihood to find the correct page at the top. In general, this trend is respected over the four criteria for all the queries groups with few exceptions, with an increasing significance when the difficulty of the query increases. The last column contains the results of a Z-test performed with a Z_{crit} level of 90%. We note how 4 out of 4 of the most difficult queries have a statistically

Table 4 Full expert function results

Query or Group	P_{suc}	N_p	N_q	T	Z-test
10	53%	12.45	4.95	8'03"	yes
15	69%	8.17	4.17	4'47"	yes
G1	86%	13.00	3.30	4'57"	yes
G2	87%	6.06	1.79	3'23"	yes
G3	93%	4.97	2.67	2'29"	no
G4	92%	4.52	2.00	2'47"	no
G5	99%	3.95	1.81	2'20"	no
G6	100%	2.89	1.05	1'20"	no

significant improvement, while there is no statistical difference in the low difficulty queries, that already show very high values for all the criteria.

6.5 Experiment 2 - No Past Performance

We wondered if the results would still remain valid by removing the past-performance factor from the expertise function. We noted that by removing this factor, by far the most used in trust models, the remaining computation can be easily implemented in an autonomic way, and therefore scalable. The results obtained are displayed in table 5. The four graphs in figure 4 display, for each criteria, the difference between baseline values and the full (gray line) and limited (dark grey line) trust function, as a percentage of the difference over the baseline values. Therefore, a positive value means that a gain is actually achieved in respect to the baseline function, while negative values mean how the function is actually under-performing the baseline value. Tables show how results are deteriorating, but there is still a significant gain for Q10 and Q15. Nevertheless, there is a significant gap between the two trust computations for the most difficult queries.

Table 5 Limited expert function results

Query or Group	P_{suc}	N_p	N_q	T
10	47%	14.65	5.95	9'12"
15	57.3%	9.17	4.2	5'21"
G1	84.2%	13.98	3.24	5'52"
G2	81.5%	7.04	1.82	3'25"
G3	92.4%	4.83	2.54	2'12"
G4	92.6%	4.78	2.10	2'56"
G5	97.9%	4.86	1.81	1'58"
G6	100%	2.73	1.10	1'28"

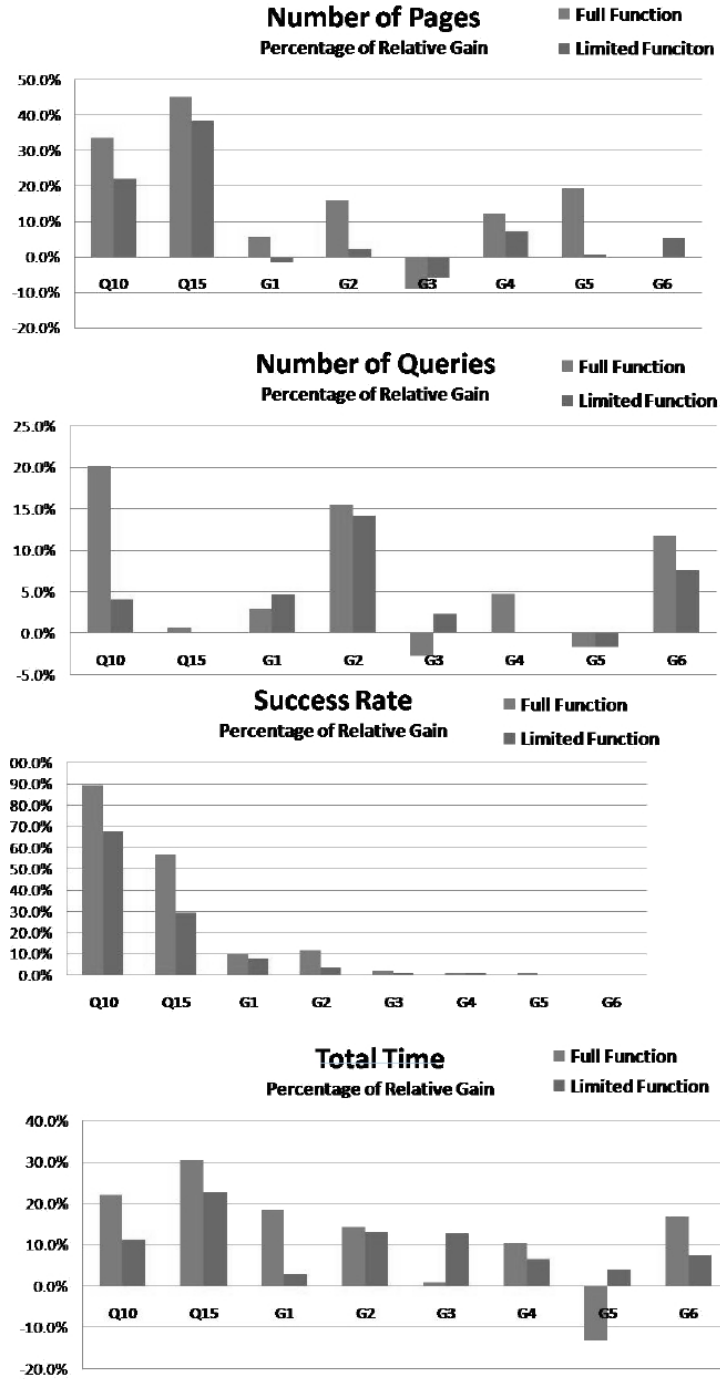


Fig. 4 Percentage of relative gain

7 Future Direction and Open Issues

With the proliferation of large-scale collaborative computing collaborative environments a new problem has emerged in the last few years: the reliability and trustworthiness of the entities involved. Trust is a form of collective intelligence used by humans as a powerful tool for the decision making process. This has enabled the proposal of computational trust models in the last decade aimed at filtering harmful entities. This chapter presented techniques to computationally assess the trustworthiness of entities involved in a collaborative context. An experimental study case was presented in the context of Social Search, where we developed and tested a trust function to spot reliable users. The experimental results show how the page ranking of the Social Search engine can be improved and the time spent to search info diminished, if enough information are collected from the community. The method proved to be more effective for difficult queries, where it is easier to get advantage of expert users. The study showed how trust techniques can improve the quality of Social Search engines, confirming their central role in deploying effective collective intelligence in the age of Global Computing.

Future work include a further expansion and enhancement of the proposed computational model by considering the effort spent by users in consuming activity over the World Wide Web [28]. This will be done in a lower-level of details, by considering the set of actions done by users while consuming the content of a particular web page, that means by considering activity such as clicking, scrolling, cut & paste, finding, bookmarking and so forth over the time dimension. This activity will be mapped to cognitive theories for cognitive effort [27] available in the fields of cognitive science, psychology, in order to further enhance the quality of predictions of a Social Search Engine by filtering not trustworthy entities based on their cognitive effort over the web.

References

1. Agichtein, E., Brill, E., Dumais, S.: Improving Web Search Ranking by Incorporating User Behavior Information. In: SIGIR 2006, Seattle, USA (2006)
2. Agichtein, E., Zheng, Z.: Identifying Best Bet Web Search Results by Mining Past User Behavior. In: KDD 2006, Philadelphia, Pennsylvania, USA (2006)
3. Atterer, R., et al.: Knowing the User's Every Move - User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In: WWW 2006, Edinburgh (May 23-26, 2006)
4. Buskens, V.: The Social Structure of Trust. *Social Networks* (20), 265–298 (1998)
5. Ball, E., Chadwick, D., Basden, A.: The Implementation of a System for evaluating Trust in a Pki Environment. In: Proceedings of Trust in the Network Economy, Evolaris (2003)
6. Celentani, M., Fudenberg, D., Levine, D.K., Pendorfer, W.: Maintaining a Reputation Against a Long-Lived Opponent. *Econometria* 64(3), 691–704 (1996)
7. Chi, E.H.: Information Seeking Can Be Social. *Computer* 42(3), 42–46 (2009)
8. Castelfranchi, C., Falcone, R.: Trust is much more than Web Probability. In: 32nd Hawaii Int. Conference (2000)
9. Cahill, V., et al.: Using Trust for Secure Collaboration in Uncertain Environments. *IEEE Pervasive Computing Magazine* 2(3), Special Issue (July-September 2003)

10. [Dondio, P., Barrett, S., Weber, S., Seigneur, J.M.: Extracting Trust from Domain Analysis: a Study on Wikipedia. In: IEEE ATC, Wuhan, China \(2006\)](#)
11. Dondio, P.: Trust as a Form of Defeasible Reasoning. Phd Thesis, Trinity College Dublin
12. Ford, N., et al.: Web Search Strategies and Human Individual Differences: Cognitive and Demographic Factors, Internet Attitudes, and Approaches. *Journal of Am. Soc. Inf. Sci. Technol.* 56, 7 (2005)
13. Gambetta, D.: Can we trust trust? . In: *Trust: Making and Breaking Cooperative Relations*, pp. 213–237 (2000)
14. [Golder, S.A., Huberman, B.A.: Usage Patterns of Collaborative Tagging Systems. Journal of Information Science 32\(2\), 198–208 \(2006\)](#)
15. Golbeck, J.: *Trust Networks on the Semantic Web*. University of Maryland, USA (2002)
16. Hume, D.: *A Treatise of Human Nature*. Clarendon Press, Oxford (1737) (1975)
17. Hlscher, C., Strube, G.: *Web Search Behavior of Internet Experts and Newbies* (2000)
18. [Josang, A., Pope, S.: Semantic Constraints for Trust Transitivity. In: 2nd Conference on Conceptual Modelling \(2005\)](#)
19. [Joachims, T.: Optimizing Search Engines Using Clickthrough Data. In: The Proceedings of SIGKDD \(2002\)](#)
20. [Karlins, M., Abelson, H.I.: Persuasion, how Opinion and Attitudes are Changed. Crosby Lockwood & Son \(1970\)](#)
21. [Kelly, D., et al.: Reading Time, Scrolling and Interaction: Exploring Implicit Sources of User Preferences for Relevance Feedback During Interactive Information Retrieval. In: SIGIR 2001, New Orleans, USA \(2001\)](#)
22. Abdi, H.: Kendall Rank Correlation. In: Salkind, N.J. (ed.) *Encyclopaedia of Measurement and Statistics*. Sage, Thousand Oaks (2007)
23. [Kitajima, M., Blackmon, M.H., Polson, P.G.: Cognitive Architecture for Website Design and Usability evaluation: Comprehension and Information Scent in Performing by Exploration. HCI, Las Vegas \(2005\)](#)
24. [Kleinberg, J.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM 46\(5\), 604–632 \(1999\)](#)
25. [Longo, L., Barrett, S., Dondio, P.: Toward Social Search: from Explicit to Implicit Collaboration to Predict Users' Interests. In: WebIST 2009 \(2009\)](#)
26. [Longo, L., Dondio, P., Barrett, S.: Temporal Factors to Evaluate Trustworthiness of Virtual Identities. In: IEEE SECOVAL 2007, Third International Workshop on the Value of Security through Collaboration, SECURECOMM 2007, Nice, France \(September 2007\)](#)
27. Longo, L., Barrett, S.: Cognitive Effort for Multi-Agent Systems. In: Yao, Y., Sun, R., Poggio, T., Liu, J., Zhong, N., Huang, J. (eds.) *BI 2010*. LNCS, vol. 6334, pp. 55–66. Springer, Heidelberg (2010)
28. [Longo, L., Dondio, P., Barrett, S.: Information Foraging Theory as a Form Of Collective Intelligence for Social Search. In: 1st International Conference on Computational Collective Intelligence Semantic Web, Social Networks & Multiagent Systems, Wroclaw, Poland, \(October 5-7, 2009\)](#)
29. [Luhmann, N.: Familiarity, Confidence, Trust: Problems and Alternatives. In: Trust: Making and Breaking Cooperative Relations](#), pp. 213–237 (2000)
30. [Marsh, S.: Formalizing Trust as Computational Concept. PhD, Stirling \(1994\)](#)
31. [Miller, C.S., Remington, R.W.: Modeling Information Navigation: implications for Information Architecture. In: HCI \(2004\)](#)
32. [Montaner, M., Lopez, B., De La Rosa, J.: Developing Trust in Recommender Agents. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems \(AAMAS 2002\), Bologna, Italy, pp. 304–305 \(2002\)](#)

33. [Morita, M., Shinoda, Y.: Information Filtering Based on User Behavior analysis and Best Match Text Retrieval. In: 17th ACM SIGIR \(1996\)](#)
34. [Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Stanford University, Stanford \(1999\)](#)
35. [Pirolli, P.: Information Foraging Theory. Adaptive Interaction with Information. Oxford University Press, Oxford \(2007\)](#)
36. [Pirolli, P., Fu, W.: SNIF-ACT: A Model of Information Foraging on the World Wide Web. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. \(eds.\) UM 2003. LNCS, vol. 2702. Springer, Heidelberg \(2003\)](#)
37. [Quercia, D.: STRUDEL: Supporting Trust in the Establishment of Peering Coalitions. In: ACM SAC 2006, pp. 1870–1874 \(2006\)](#)
38. [Robu, V., Halpin, H., Shepherd, H.: Emergence of Consensus and Shared Vocabularies in Collaborative Tagging Systems. ACM Transactions on the Web \(TWeb\) 3\(4\), article 14 \(September 2009\)](#)
39. [Stephens, D.W., Krebs, J.R.: Foraging Theory, Princeton, NJ \(1986\)](#)
40. [Sabater, J., Sierra, C.: REGRET: A reputation Model for Gregarious Societies. In: 4th Workshop on Fraud and Trust in Agent Societies, Montreal, Canada, pp. 61–69 \(2001\)](#)
41. [Velayathan, G., Yamada, S.: Behavior-based Web Page Evaluation. In: WWW 2007, Banff, Alberta, Canada, May 8-12 \(2007\)](#)
42. [Viégas, B.F., Wattenberg, M., Kushal, D.: Studying Cooperation and Conflict between Authors with History from Visualizations, MIT Media Lab. and IBM Research](#)
43. [Weiss, A.: The Power of Collective Intelligence. Collective Intelligence, 19–23 \(2005\)](#)

