Context and Action in Search Interfaces

Alan Dix[1,2]

[1] Talis, Solihull Parkway Birmingham Business Park, Birmingham B37 7YB, UK
[2] Lancaster University, School of Computing and Communications, Lancaster LA1 4WA UK
alan@hcibook.com

Abstract. While the web is often described in terms of access to information, it is also a place where people *do* things from booking hotel rooms, to completing their tax return. This paper outlines the ways in which search can form a part of a more action-based view of web interaction. The simplest is that search can be action that the user is performing to get information. However, search can also be used more computationally within an intelligent system that infers appropriate points to trigger interaction (loci of action) and constructs a model of the users context. The resulting picture is a rich interplay between user action and computation, where each inform and influence the other, and where search can form an intimate part both explicitly for the user and embedded within computation.

Keywords: user interaction, context inference, data detector, intelligent user interface.

1 Overview/Motivation

While the web is often described in terms of access to information, it is also a place where people *do* things from booking hotel rooms, to completing their tax return [1]. Theories of embodiment stress that, as creatures *in the world*, our perception is not an abstract gathering of information for processing, but an integral part of our being acting beings [2]. This is equally true in the digital world of the web: we search for information to do things whether on the web, in the world or both; for example, looking at potential holiday destinations, which will later be booked online but visited physically. Even information seeking is an interactive process as made clear in information foraging theory [3], where information we have allows us to make choices of actions (including performing more searches) which lead to further information. This exactly parallels the ecological understanding of perception, where we may turn our head or move to see things better.

Search is thus an integral part of this acting view of human activity on the web, both informing human actions and being one of those actions. If we regard web interaction as solely instrumental: where the user is the only actor and computational elements are tools or data, then that is the end of the story. However, if we allow the computation to take a more active role as mediator or assistant, then we can see a wider role.

S. Ceri and M. Brambilla (Eds.): Search Computing II, LNCS 6585, pp. 35–45, 2011.

For a human listener a request such as "where is the bus" might be interpreted "when is the bus coming, is it late" if said at a bus stop, or "where is the bus stop" if said in a shopping centre. If you know someone well then when they say "RDF" you will know whether they mean "Resource Description Framework" or "Refuse Derived Fuels". When human helpers hear a query or request they will use *context* in order to work out the likely intention. This context may include the nature of the person as well as the situation in which the request occurs. Similarly, context such as personal profile information or the recent activity of a user can be used to interpret the user's actions and information they have been viewing and use this to better respond to the user's future actions and requests.

Furthermore, if you meet a visitor in your hometown and they mention they like art, you may suggest a gallery or local attraction based on that, even *without being asked*. If the person then showed interest you may then go on to suggest the best way to get there, walking, by bus, again without being explicitly asked. That is the human helper may notice *loci for action*, topics or things in the conversation or previous activity that suggest that information or potential actions may be appropriate. In a computational setting this locus may be a phrase in an email or web page being visited, or something in the output of a previous action, for example the destination town after having booked a rail ticket.

As previously noted, search may be a potential action, which may have been suggested based on a locus for action (e.g. searching for information on the destination town), and that search may be influenced by context (e.g. if the user frequently looks up art-related material). However, search can also be used computationally as part of the algorithmic processes that infer potential loci for action and user context.

In the next section we will examine the cycle of digital interaction, seeing both the role that loci-for action and context can play in this and also the way that search can be an enabler and outcome of the cycle. In Section 3 and Section 4, we look in turn at how loci for action and context can be inferred; in each case seeing how search can be used computationally. Section 5 then looks at search as an action itself, but the way in which the loci and context can influence the search process and in particular the choice of search repository.

2 Cycles of Action and Context

Fig. 1 shows the cycle of user action and the role automated context can play in this cycle. The user is engaged in a series of actions: on the web these would typically be clicking a link, hitting a form button, typing in a url, or entering a search string. The actions each create outputs; for example a page of search result, a web page, email message, PDF or Word document (the term 'document' will be used generically to refer to all these).

Sometimes the user may spontaneously invoke an action (e.g. typing a url), but very often actions arise out of the results of previous actions (e.g. clicking a link, reading an email message, opening a document); that is a loci of action, as introduced in the previous section. In some cases the potential loci will be very explicit (link or button); however, in others there may be something in the output document (e.g. a person's name) that may be the trigger for the user's next action, but which is not obviously encoded as such. The user will typically have many choices for a potential next action, and so there is a cycle where the user performs an action, obtains results and may select some part of this output for the next action.

Both the detection of these more implicit loci for action and the execution of the action may be influenced by automatically inferred context based on the user's previous interaction history both long term (building up a broad profile of the user's preferences) and shorter term (understanding the user's current topic of interest).

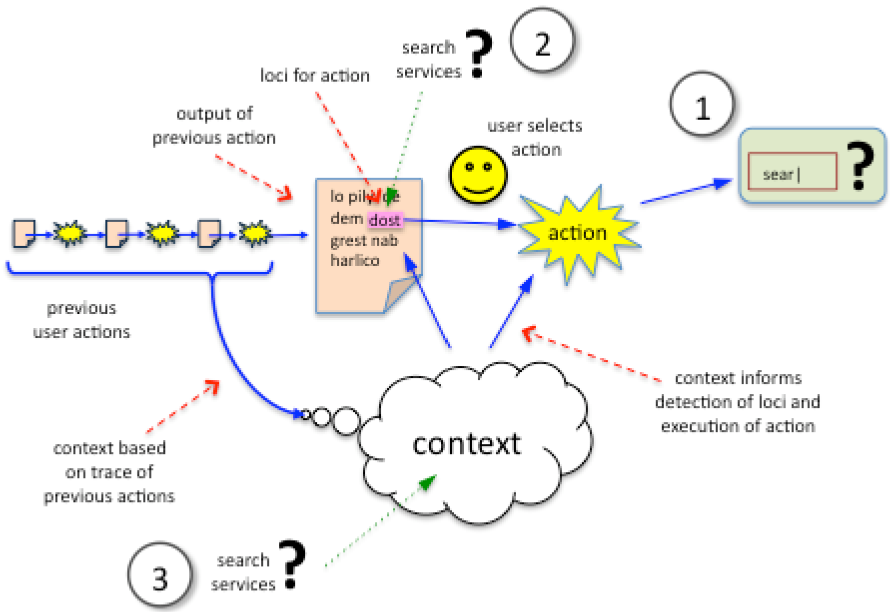

Fig. 1. Cycle of action and context

Search may be involved in different parts of this cycle:

1. the action being invoked may be a search, in which case the loci and/or automatic context may be used to help select appropriate search services, or influence ranking
2. search-based resources can be used together with context in determining loci for action, whether in an web page, email message or document.
3. context inference may use search-based resources in order to enrich trace data

3 Establishing Loci for Action

In some cases the document that the user is viewing may have explicitly encoded loci for further action such as a hyperlink or button. In these cases the application or person that produced the document being viewed determines the next action. However, more interesting are implicit loci, such as names, or places mentioned in text, as these may be used in ways not foreseen by the originator: one user may use a name as input into Google Scholar, another for a Facebook search, and yet another as input into IMDb. If one can determine which parts of the document constitute potential loci and furthermore what kind of data each represents (e.g. person name, place name, telephone number), then it is possible to guide the user towards suitable resources and actions including the choice of appropriate search services.

3.1 Explicit Semantic Markup of Loci

One way in which these loci can be detected is when the document originator has added explicit semantic mark-up. While still not widespread, various forms of mark-up including micro-formats and RDFa are beginning to be used on the web; for example, Figure 2 shows a fragment of the HTML of a LinkedIn profile page where hCard microformat [4] encoding of vCard [5] is used to mark the words "Alan Dix" as a name where "Alan" is the given name and "Dix" is the family name.

```
<div class="masthead vcard contact">
  <div id="nameplate">
    <h1 id="name">
        <span class="fn n">
            <span class="given-name">Alan</span>
            <span class="family-name">Dix</span>
        </span>
    </h1>
  </div>
```

Fig. 2. hCard microformat on profile page at linkedin.com

Of course, the vast majority of web resources are not marked up in this way, indeed the main reason that the LinkedIn profile can be annotated like this is because it is being generated from a database where the semantics of the name are already explicit. If, for example, the name were mentioned in the middle of an email message from a friend it is unlikely that the writer of the email would explicitly mark-up the content in this way!

3.2 Inferring Loci – Data Detectors

Happily it is possible to automatically detect this type of semantic value within text using a form of local text mining known as data detectors. Data detectors have been used sporadically since the late 1990s when a number of commercial and research systems were developed including work at Intel [6], Apple [7], Georgia Tech (CyberDesk) [8] and aQtive (onCue) [9]. Some of these worked by scanning the complete text available to the user, others some selected portion; for example, onCue worked by looking at the clipboard so that it could react whenever the user copied or cut text. All of these early systems effectively used syntactic methods for recognition, using some variant of BNF, regular expressions, or bespoke code to analyse the text and work out what it represented from patterns of letter use, punctuation etc. For example a UK postcode matches the regular expression below:

```
/([A-Za-z][A-Za-z0-9]{1,3})[ \t]*([0-9][A-Za-z]{2})/
```

Of course, as with any such system, this can lead to both false positives (e.g. "High Court" is classified as a person name because of its length and capitalisation) and false negatives (e.g. the poet name "e.e. cummings" is not recognised because of its unusual capitalisation). Because of this, onCue was deliberately designed using principles of 'appropriate intelligence', embedding the 'intelligent' algorithms within an interaction framework that meant errors in the intelligence did not negatively impact the user experience [9].

3.3 Using Search in Data Detectors

In even earlier work, the Microcosm hypermedia system developed at Southampton University [10] added links dynamically at the server side using a similar form of text mining. However, instead of syntactic rules Microcosm looked up the terms in the source document and matched them against a database linking keywords/terms to particular resources. This is a similar to the way terms link to their topics within Wikipedia by keywords/phrases not full hyperlinks; except that in Microcosm there was no need for *any* explicit mark-up by the page author, not even to say "this a term". Note that this is effectively using *search* in order to deliver 'intelligent' results, in the same way that Google 'suggest' does while typing search terms.

This form of search-based data detection and the more syntactic rules used by most data detectors can be combined. The web-based bookmarking system Snip!t (www.snipit.org) uses technology developed from onCue in order to perform data detection over selected fragments of web pages; Figure 3 shows Snip!t suggesting potential actions after recognising a post code in a selected web page fragment. Snip!t uses some plain syntactic rules like onCue and the other early data detectors, but also performs semantic searches in larger tables of data like Microcosm. These can be used separately for different kinds of data, regular expressions for postcodes, lookup for city names. However, most powerfully then two can be combined.

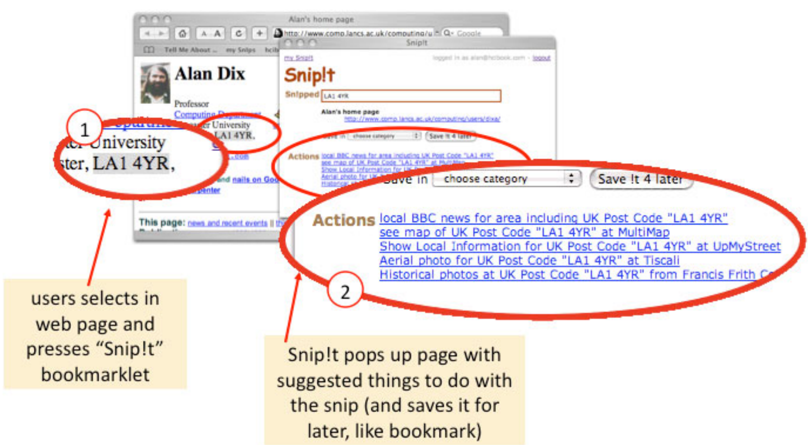

Fig. 3. Snip!t recognises a post code and suggest potential actions

An example of this can be seen in Figure 4, which shows a portion of the XML description file for a person's name. Towards the top it declares "<keys>" and lists "Female First Name" and " Male First Name". These are the names of data types that have already been recognised using a lookup in a large table of male and female first names compiled from US census data. The rest of the XML description then gives syntactic patterns for what could appear before or after this first name in order to form a complete name. While not eliminating false positives and false negatives, the use of lookup/search does increase the accuracy of the recogniser and is also more efficient as the syntactic matching rule is only invoked if a suitable lookup match is found.

```
<beforeafterrecogniser>
  <name>name1Recogniser</name>
  <title>Person Name recogniser</title>
  <keyed>
    <keys>Female First Name, Male First Name</keys>
  </keyed>
  <pattern>
    <pre_context>\W+</pre_context>
    <before>($RE_HONOURIFICS\s*)?</before>
    <key>$RE_CAPSNAME</key>
    <after>\s*$RE_MIDNAMES\s*$RE_LASTNAME</after>
    <post_context>\W+</post_context>
  </pattern>
  <match>
    <type>name</type>
    <description>Person name $$</description>
  </match>
</beforeafterrecogniser>
```

Fig. 4. Snip!t recogniser description file keyed from name lookup

4 Establishing Context

The techniques used to establish loci can also be used as a form of immediate interaction context. If you visit a web page that mentions Milan, it is likely that Milan itself, Italy in general, or maybe research groups within Milan may be your topic of interest. However, when looking at past activity other forms of inference can be used.

Most obviously, the order of past actions can be used to suggest potential future actions, for example, using hidden Markov models or other forms of sequence inference. This can be a powerful technique, but is not strongly related to search computing, so is not discussed further here, but more details can be found in [1].

More pertinently, the input of users can be analysed in a similar way to the outputs of actions. These user inputs are often more likely to refer to data of immediate personal importance and this can be used to yield more precise results. For example, if the term "Tiziana" appears on a web page, it is possible to work out this is a female first name, but it could refer to many Tiziana's. However, if you type

"Tiziana" into an input field, then it is likely that the Tiziana in question is someone you know and local resources, such as an address book, can be searched to determine that this is very likely to be "Tiziana Catarci".

Various projects, including those looking at the 'Semantic Desktop' [11,12] and the TIM project [13,1], consider some form of personal ontology, where personal constructs are linked together, for example explicitly recording that "Tiziana Catarci" 'works in' "University of Roma, La Sapenza", where each of the terms is a semantic entity or relation in the ontology. This data may be mined from existing sources such as address books, calendars and email messages, or entered explicitly using a dedicated interface.

Based on similar knowledge to that embedded in these personal ontologies, an experienced (human) personal assistant would be able to tell that if you had just received an email from Tiziana and then asked for a flight and hotel to be booked, then Rome is a likely potential location, and if so then the hotel needs to be close to "La Sapienza". If a user has a rich digital personal ontology then it becomes possible for a computer to make similar inferences.

In the TIM project and related work we have looked at both rule-based inference of relationships between form fields and also more 'fuzzy' inference using spreading activation. In the case of more precise rules it is possible to infer that if a form has both a name and address field, that the address is likely to be the one associated with the person, and moreover, based on previous form entry, whether it should be the home or business address [1].

Spreading activation is used to suggest potential linkage even where there has been no previous exposure to the precise forms or applications [14]. For example, if an email has been received from Vivi, then the entity representing Vivi is made very 'active' in the ontology; then other entities in the ontology that have close associations with Vivi (maybe a colleague Costas, or her institution 'University of Athens') are also given some activation (see figure 5). In turn entities further and further from Vivi inherit smaller amounts of activation from their neighbours. In this way those entities in the ontology, which are closer to the initially activated entity are most active and this can be used to rank potential candidates to complete future parametrised actions.

This works fine so long as all the relationships are in the personal ontology, but of course a human assistant would know, for example, that Athens is in Greece even if this were not explicitly stated in an address book. Basic spreading activation is limited to the knowledge available, but happily further information is available on the web in both semantic-web linked-data sources [15] and also searchable deep-web search services [16]. It is thus possible to fill the gaps in the locally stored explicit knowledge with this globally stored, often 'common sense' knowledge, for example, knowing (from the personal ontology schema) that 'Athens' is a place, then if Athens is highly activated one can search resources such as the Geonames online database to find information about Athens, which will include that it is the capital of Greece. Initial studies have shown that this web-scale reasoning using spreading activation is feasible and scalable [17].

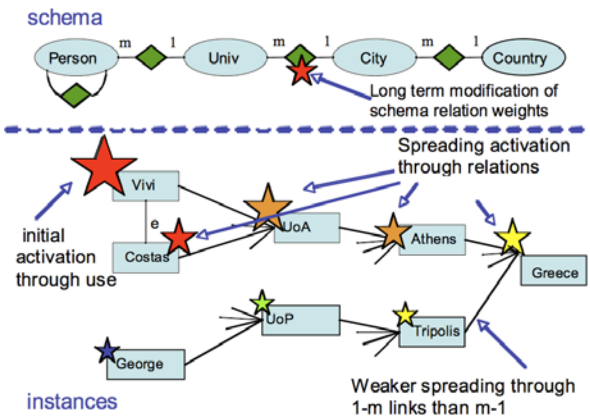

Fig. 5. Spreading activation in a personal ontology

5 Suggesting and Influencing Search

As noted search may be one of the potential actions. If this is the result of a locus for action from a data detector, then the semantics of the identified term can be used to tune the search. For example, Snip!t suggests directory search services such as ZoomInfo if the term is a person name, location-based search services such as recycle-more.co.uk if the term is a post code, and search at acronymfinder.com if the term appears to be an acronym. This already gives a high level of specificity, however, there are some types of term for which there are many possible search services, or the term may be ambiguous. In such case the slightly broader context given by spreading activation may allow greater specificity.

So far this has not been implemented with Snip!t, but the author used somewhat similar techniques during in research at aQtive in the late 1990s. The Open Directory Project was used which has both a hierarchical category system and a large corpus of web material hand-classified in the hierarchy. For any term the relative frequency of the word in pages associated with leaf categories can be used to give an initial 'activation' of the leaf category, which can then be spread both up and down the category hierarchy in order to generate a relevance measure of a term for a category, even if there is no explicit mention of the term in the category. This is a constrained form of spreading activation, as it is limited to a hierarchy, but otherwise very similar to spreading activation on the personal ontology.

The outputs of this process were then used to classify unseen text; for example the term "Chihuahua puppy" would be correctly classified with highest confidence as Chihuahua the dog rather than Chihuahua the place in Mexico even if the term 'puppy' never appeared in pages classified as "pet/dog/chihuahua", because pages in other subcategories of the dog category did mention 'puppy'. This was then used to improve the relevance results of searches of various kinds: web pages, images and online shops. This information could also tune the choice of appropriate search resources (e.g. zoominfo, or IMDb to lookup a name) [18].

6 Discussion

We have seen a number of ways in which search computing interacts with a more action and context focused view of user interaction on the web.

On the one hand search can be used to enhance user interaction by working together with more syntactic methods to allow data detectors to identify loci for actions and more generally text mining to establish potential semantic entities. Similarly when spreading activation is used for context inference, local data stores can be augmented on the fly by external data sources including searches of deep web resources. The appropriate points for augmentation can be identified based on 'chasing' already loaded entities of high activation, and the appropriate deep web resources can be indentified because the personal ontology establishes a clear type for the data.

In the above, search computing was used as a hidden part of the algorithms used to suggest potential future actions, establish user context and based on this aid in the completion of data for actions. However, search computing can also be the final point of such actions; that is the above techniques may yield search as a final action of the user. In these case the fact that loci of action can be identified and given a precise type can help choose or rank appropriate search repositories and moreover this can be tuned further by inferred user context.

More broadly it is interesting to note the way in which user interaction is not only enabled by existing knowledge such as the personal ontology and search data, but is also the source of such knowledge in terms of traces of interaction. It is easy to conceptualise the role of algorithmics in user interaction in terms of the computation that occurs between user actions, that is no more than a variation of a 1960s batch input-output process. However, interactive computation [19] has different properties to plain IO computation, and we need to re-conceptualise with a more *systemic* view where the human activity is integral to the computation, as has been emphasised in recent work on 'human computation' [20, 21].

Knowing that a human will interpret results may change our focus. Rather than seeing algorithms as producing precisely the best the results, we focus more on exposing alternatives and rationale. Similarly knowing that users are acting on outputs of one phase of an algorithm may mean that user actions implicitly feed information back into the automated analysis. For example, faced with the digits "0152410317" in text, data detectors may offer multiple suggestions, not just the 'best' one say as a phone number or a 10 digit ISBN; however if the user selects "phone using Skype" this is a confirmation that it is the phone number not the ISBN.

When we view the computer in this more interactional role, search becomes a natural partner for more traditional algorithmics. In human cognition the relatively slow process of sequential rational thought compliments the massively parallel associative access to past memories. The vast quantities of data available on the web are proving surprisingly powerful [22,23] as a computational resource, and in this paper we have seen examples of how search acting as an associative store compliments more explicit, rule-based algorithms in a similar way to our own human thinking. Search is therefore not only a way to provide information for the human user, but has the potential to allow the computer to act in a more humanly comprehensible way.

References

1. Dix, A., Lepouras, G., Katifori, A., Vassilakis, C., Catarci, T., Poggi, A., Ioannidis, Y., Mora, M., Daradimos, I., Md. Akim, N., Humayoun, S.R., Terella, F.: From the Web of Data to a World of Action. Web Semantics: Science, Services and Agents on the World Wide Web 8(4), 394–408 (2010), doi:10.1016/j.websem.2010.04.007.
2. Clark, A.: Being There: Putting Brain, Body and the World Together Again. MIT Press, Cambridge (1998)
3. Pirolli, P.L.: Information foraging theory: adaptive interaction with information. Oxford University Press, Oxford (2007)
4. Çelik, T., Suda B.: hCard 1.0., `http://microformats.org/wiki/hcard` (accessed December 29, 2010)
5. Dawson, F., Howes, T.: vCard MIME Directory Profile. RFC 2426, IETF (1998), `http://www.ietf.org/rfc/rfc2426.txt`
6. Pandit, M., Kalbag, S.: The selection recognition agent: Instant access to relevant information and operations. In: Proc. of Intelligent User Interfaces (IUI 1997), pp. 47–52. ACM Press, New York (1997)
7. Nardi, B., Miller, J., Wright, D.: Collaborative, Programmable Intelligent Agents. Communications of the ACM 41(3), 96–104 (1998)
8. Wood, A., Dey, A., Abowd, G.: Cyberdesk: Automated Integration of Desktop and Network Services. In: Proc. of the Conference on Human Factors in Computing Systems (CHI 1997), pp. 552–553. ACM Press, New York (1997)
9. Dix, A., Beale, R., Wood, A.: Architectures to make Simple Visualisations using Simple Systems. In: Proceedings of Advanced Visual Interfaces, AVI 2000, pp. 51–60. ACM Press, New York (2000)
10. Hall, W., Davis, H., Hutchings, G.: Rethinking Hypermedia: The Microcosm Approach. Kluwer Academic Publishers, Norwell (1996)
11. Semantic Desktop. semanticweb.org, `http://semanticweb.org/wiki/Semantic_Desktop` (accessed October 2010)
12. Sauermann, L., van Elst, L., Dengel, A.: PIMO - a framework for representing personal information models. In: Pellegrini, T., Schaffert, S. (eds.) Proceedings of I-MEDIA 2007 and I-SEMANTICS 2007 as part of TRIPLE-I 2007, pp. 270–277 (2007)
13. Catarci, T., Dix, A., Katifori, A., Lepouras, G., Poggi, A.: Task-Centred Information Management. In: Thanos, C., Borri, F., Candela, L. (eds.) Digital Libraries: Research and Development. LNCS, vol. 4877, pp. 197–206. Springer, Heidelberg (2007)
14. Katifori, A., Vassilakis, C., Dix, A.: Ontologies and the Brain: Using Spreading Activation through Ontologies to Support Personal Interaction. Cognitive Systems Research 11, 25–41 (2010)
15. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems, Special Issue on Linked Data 5(3), 1–22 (2009)
16. Sherman, C., Price, G.: The Invisible Web: Finding Hidden Internet Resources Search Engines Can't See. CyberAge Books (2001)
17. Dix, A., Katifori, A., Lepouras, G., Vassilakis, C., Shabir, N.: Spreading Activation Over Ontology-Based Resources: From Personal Context To Web Scale Reasoning. Internatonal Journal of Semantic Computing, Special Issue on Web Scale Reasoning: scalable, tolerant and dynamic 4(1), 59–102 (2010)

18. Dix, A., Karapetis, F., Kefalidou, G.: Concept Classification for Decentralised Search. Working Paper, Lancaster University, November 11 (2005), http://www.hcibook.com/alan/papers/Concept-Classification-for-Decentralised-Search-2005/
19. Wegner, P., Goldin, D.: Computation beyond turing machines. Commun. ACM 46(4), 100–102 (2003), DOI= http://doi.acm.org/10.1145/641205.641235
20. von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science 321, 1465–1468 (2008)
21. Eagle, N.: txteagle: Mobile Crowdsourcing. In: Aykin, N. (ed.) IDGD 2009. LNCS, vol. 5623, pp. 447–456. Springer, Heidelberg (2009)
22. Halevy, A., Norvig, P., Pereira, F.: The Unreasonable Effectiveness of Data. IEEE Intelligent Systems 24(2), 8–12 (2009)
23. Finin, T., Syed, Z., Mulwad, V., Joshi, A.: Exploiting a Web of Semantic Data for Interpreting Tables. In: Proceedings of the WebSci 2010: Extending the Frontiers of Society On-Line (2010)