International Journal of Semantic Computing
© World Scientific Publishing Company

SPREADING ACTIVATION OVER ONTOLOGY-BASED RESOURCES FROM PERSONAL CONTEXT TO WEB SCALE REASONING

ALAN DIX

Computing Department, InfoLab21 Lancaster University,
Lancaster, LA1 4WA, UK
alan@hcibook.com
http://www.hcibook.com/alan/

AKRIVI KATIFORI

Department of Informatics & Telecommunications, University of Athens,
Athens, Hellas (Greece)
vivi@di.uoa.gr

GIORGOS LEPOURAS

Dept. of Computer Science and Technology, University of Peloponnese,
Tripolis, Hellas (Greece)
gl@uop.gr

COSTAS VASSILAKIS

Dept. of Computer Science and Technology, University of Peloponnese,
Tripolis, Hellas (Greece)
costas@uop.gr

NADEEM SHABIR

Talis, Birmingham, UK,
nadeem.shabir@talis.com

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

This paper describes methods to allow spreading activation to be used on web-scale information resources. Existing work has shown that spreading activation can be used to model context over small personal ontologies, which can be used to assist in various user activities, for example, in auto-completing web forms. This previous work is extended and methods are developed by which large external repositories, including corporate information and the web, can be linked to the user's personal ontology and thus allow automated assistance that is able to draw on the entire web of data. The basic idea is augment the personal ontology with cached data from external repositories, where the choice of what data to fetch or discard is related to the level of activation of entities already in the personal ontology or cached data. This relies on the assumption that the working set of highly active entities is relatively small; empirical results are presented, which suggest these assumptions are likely to hold. Implications of the techniques are discussed for user interaction and for the social web. In addition, warm world reasoning is proposed, applying rule-based reasoning over activate entities, potentially merging symbolic and sub-symbolic reasoning over web-scale knowledge bases.

Keywords: personal ontology; spreading activation; web-scale reasoning; context modelling intelligent user interface; personal information management, warm-world assumption

1. Introduction

A typical computer user may have hundreds of address book contacts, many thousands of files, and probably tens of thousands of emails. However, having this wealth of data stored does not help users in performing their tasks with the computer unless it is available when needed. Often this involves navigating one's way round a file system, or searching email folders for an elusive address or telephone number, or simply retyping information that you know is there somewhere.

The aim of the research that gave rise to this paper is to support users so that they have the right information available at the right time. Ideally the computer should be able to perform like an efficient personal assistant combining computer-like power and memory with the human-like understanding of the individual and the context. For example, if a user is filling out a web form, then an 'address' field would by default pre-fill to their own address, but if they have just had an email from a friend, then we might expect the friend's address to be suggested alongside or ahead of the user's own address.

One of the techniques we have been using to tackle this is spreading activation over personal ontologies [1]. In short, this involves taking a populated ontology of the user's personal information including personal profile, relationships to colleagues, projects, etc., and then applying a spreading activation algorithm over the nodes. In the above example, the email from the friend would initially excite the node in the personal ontology representing the friend; this would then spread some activation to neighboring nodes so that the friend's address would also become 'hot'. Later when the user comes to the web form the 'hottest' address in the populated ontology would be presented first as an option for the 'address' field, which would be the friend's address as required.

Our initial work in this area has proved promising and is reported elsewhere [1], but has been restricted to information held within the user's personal ontology. However, not all of the relevant information will be stored locally.

Imagine if the data in the personal ontology simply contained 'Person("Akrivi") lives_in City("Athens")'. This would be fine if the web form asked for a city, but if the required field was 'country', it would have no contextual suggestions and would simply have to default to the user's own country. A human assistant at this point would simply use their general knowledge and suggest "Greece", or, if the town or city was less familiar maybe Google it.

An automated system in principle may have the entire web available and in particular the web of 'linked data' [2,3] (interlinked computer-readable information based on semantic-web technology), so may be able to make use of 'general knowledge' and eternal data in the same way a human might. This raises the question as to whether the kinds of reasoning we have applied to personal ontologies can be extended to the entire web without needing to suck the whole of the web into a single machine.

This paper presents methods to scale spreading activation to allow web-based reasoning, based on dynamically identifying a relatively small, but appropriate, 'working set' of entities and relations. We have not as yet integrated these algorithms into our inference system, but instead present data from our own system and the literature to validate the assumptions on which our scaling algorithms are based.

While other proposals for web-scale reasoning are focused on symbolic reasoning (e.g. [4,5,6]), we are adopting neural-inspired sub-symbolic processing, in the sense that in the presented algorithm ontology concepts and relationships between concepts can be considered analogous to neurons and synapses, respectively. Given the success of Google page rank, there is *prima facie* case for the efficacy of these kinds of algorithms. However, with web-scale reasoning, many of the distinctions between symbolic and sub-symbolic reasoning begin to break down; once we hit web scale even symbolic reasoning may have to become approximate and defeasible [4]. We propose ways in which our use of spreading activation can be combined with more symbolic reasoning. By only performing the symbolic reasoning over sufficiently 'activated' information, we allow bounded reasoning within unbounded data, in a manner similar to human reasoning. We call this the *warm world assumption*.

The next section reviews a number of key concepts with relevant literature: personal ontologies, task-based interaction, spreading activation, sources of web data and web-scale reasoning. Section 3 describes the current implementation of spreading activation over a personal ontology. Section 4 introduces methods to extend spreading activation to the web and the following section presents empirical data supporting the key assumptions. Finally, Section 6 discusses a number of issues raised and in particular the means by which web-scale spreading activation can be combined with symbolic rules to give warm-world assumption reasoning.

2. Background and Concepts

In this section we review relevant concepts from the literature. We begin with *personal ontologies*, which are state-of-the-art tool for modelling and reasoning over personal context. Then we briefly discuss *task-based interaction*, which is a very active research topic exploiting personal context and has been used as a proof-of-concept application for our work, and *spreading activation*, which has provided the inspiration for the main algorithm presented in the paper. Finally, we overview the sources of web data, which can be used to enrich personal ontologies, promoting the latter from *local-scale* to *web-scale*. We conclude this section with an overview of other web-scale reasoning approaches.

2.1. *Personal Ontologies*

According to [7], an ontology is an explicit specification of a conceptualization. The term "conceptualization" is defined as an abstract, simplified view of the world that needs to be represented for some purpose. It contains the concepts (classes) and their instantiations (instances) that are presumed to exist in some area of interest and their properties and relations that link them (slots).

Using an ontology to model semantics related to the user personal domain has already been proposed for various applications like web search ([8,9]). Most of these approaches use ontologies only as concept hierarchies, like hierarchies of user interests, without particular semantic complexity, as opposed to our approach which incorporates the full range of ontology characteristics.

The value of ontologies for personal information management has also been recognized and there is on-going research on incorporating them in PIM (Personal Information Management) systems like OntoPIM [10], GNOWSIS [11] and the semantic desktop search environment proposed in [12]. However there are very few detailed works available on the exact personal ontology to be used for such an application.

The Personal Ontology used in our work constitutes an extended and enriched version of a user profile maintained by most applications as it attempts to group under one structure the user personal information, contacts, interests, important events, etc.

More details on the creation of the personal ontology may be found in [13] and [14]. The ontology, along with example instances may be found in [15].

The personal ontology attempts to encompass a wide range of user characteristics, including personal information as well as relations to other people, preferences and interests. To be as complete as possible the ontology has drawn on existing de facto standards such as FOAF and vCard as well as proprietary profiles such as Facebook. However, we do not expect this to be final or complete, so we foresee evolution of the base ontology, but more important the ontology may be extended through inheritance and the addition of more classes, as well as class instantiation according to the needs of both user stereotypes or individuals.

The addition of weights on classes, instances and relations has been the final step to make the personal ontology ready for use and testing within our spreading activation framework.

2.2. *Task-based Interaction*

While personal ontologies can help users organize and manage their information, in everyday interaction a user is not directly concerned with the information management but rather she is interested in performing tasks. To this end, user interaction support should be structured around the tasks a user executes. As illustrated in [16], in order to perform a task a user carries out certain action(s) using data related to the task's context; in this work, context is about "What to Do and What to Do It to", including thus the task that the user is involved in (e.g. reading a mail, filling in a web form) and the data involved in the task (e.g. the e-mail sender, entities referenced in the mail message body or fields and field values in the web form). Although most actions are performed on the user's own PC (albeit some being functions offered by locally installed application and some from web-based applications), data can come from a variety of sources, including user-owned devices such as PC, PDA and mobile phone or web-stored information.

Currently, to achieve their goals users resort to searching for, retrieving, copying and pasting the necessary information between applications and locations. In web-based applications, browsers can perform a level of automatic form filling using a combination of URLs and named fields. Research systems, including that of the Simplicity project [17] and W3C draft "Client Side Automated Form Entry" [18], have extended this to include mappings between specific form's field names and user profile data.

Our own work has gone beyond automatically filling-in fields by name or basic types; in related work with colleagues, we have shown how rich ontological type tags such as "name_of Friend" can be automatically inferred over an unconstrained personal ontology and furthermore how they can be linked across a single form, or multiple forms in subsequent interactions. For example, if a form contains both name and city, then after a single example this may be automatically tagged as "name_of Person p" "location_of Institution employing Person p", connecting the two fields, so that when the name is filled the location can be auto-completed [16,19]. This is useful when there is a functional relationship between fields (e.g. address of a person already entered), but does not help with a first empty form (whose name?) or where there are alternatives (home address or work address).

To offer the appropriate data during the user's interaction, the system has both to identify user actions as they carry out their tasks and to understand the context of the actions. The "what to do" part of the context, the fact that you are in the middle, say of booking a hotel room, is tackled by sequence/task inference techniques described elsewhere [16]. In this paper, we are interested in the "What to do it to" part, the initial name field or the choice between alternatives. For this we employ spreading activation (as described in section 3) as a means to predict context of actions and present via a drill-down technique the relevant data and possible actions that can be performed upon the data.

2.3. *Spreading Activation*

Spreading activation was first proposed as a model of cognition [20], but is not a new concept in semantic networks related computational research, where there are a number of proposed applications of spreading activation, especially in the area of information retrieval [21].

Crestani [22] proposes the use of spreading activation on automatically constructed hypertext networks in order to support browsing within these networks; in this case, constrained spreading activation is used in order to avoid spreading through the whole network. The work in [22] presupposes that semantics and weights have been assigned to the links within the hypertext network, possibly in an automatic/semi-automatic fashion, this however is infeasible at web scale. Liu et al [23] use spreading activation on a semantic network of automatically extracted concepts in order to identify suitable candidates for expanding a specific domain ontology. Xue et al [24] propose a mining algorithm to improve web search performance by utilizing user click-through data. Weighted relations between user queries and selected web pages are created and spreading activation is performed on the resulting network in order to re-rank the search results of a specific query, allowing also faster incorporation of newly generated pages into search results by building *similarity sets*. While this approach may improve the efficiency of searches, it offers only results at document granularity, whereas in a number of applications, including task-based interaction, entity-level granularity is far more useful. Besides, only terms appearing in user queries are considered, which do not necessarily cover the full breadth of ontological resources, especially when the scope of the application is a single user's interaction.

Hasan [25] proposes an indexing structure and navigational interface which integrates an ontology-driven knowledge-base with statistically derived indexing parameters, and the experts' feedback into a single spreading activation framework to harness knowledge from heterogeneous knowledge assets. While the authors mention the existence of a local learning rule which modifies the weight on links directly or indirectly involved in a query process, no further details are provided for this; moreover, in [25] expert need to provide direct feedback for adapting the network weights, and no method for linking to external (web) sources is provided. Finally, the discussion on scalability is limited to how new documents can be incorporated to the system, implying that all information can be hosted in a single computer.

It is also worth noting that although the works [22], [24] and [25] consider spreading activation, they do not deal with the different timescales of memory. [22] refers to "some form of activation decay" that may be included in the (optional) preadjustment or postadjustment phases; [24] includes a decay factor; and [25] includes an activation retention threshold for the same purposes. However, these provisions only model how importance of items is lost, and do not capture the notion of the "current task".

Neural networks and in particular Hopfield Networks [26] attempt to approach and simulate the associative memory again by using weighted nodes but at a different level. In this case, the individual network nodes are not separate concepts by themselves, but rather, in their whole, are used to represent memory states. This approach corresponds to the neuron functions of the human brain and mainly focuses on the storage of memories, whereas ours attempts to simulate the human memory conceptual network functions and focuses on the representation of activation of individual concepts.

Recently, spreading activation theory has been recognized as a candidate approach for supporting personal interaction with the system, in the newly emerging areas of personal information management (PIM) and Task Information Management (TIM). This work has been published in [27] and [1] and is summarized in section 3.

2.4. *Sources of Web Data: Linked Data and the Semantic Web*

In the scenario in Section 1, the human assistant would either just 'know' that Athens is in Greece, or, if not, Google "Athens" to find out. Of course while the web is full of human-readable information, much of this is unavailable for automated reasoning. The goal of the Semantic Web is to change this [28] and make a 'web of data'.

While some use of semantic web technology is effectively still in vendor specific ‘silos’ either private or using bespoke ontologies, there is a growing body of ‘Linked Data’ [2,3], that is web services that use semantic web technology (RDF, SPARQL, usually REST-ful), but also use interlinked ontologies so that entities in one can be linked to those in another. Figure 1 shows some of these sources, for example DBpedia, which extracts the data in Wikipedia ‘info boxes’ and turns it into RDF data, and Geonames, which does the same for geographic information from a number of sources.

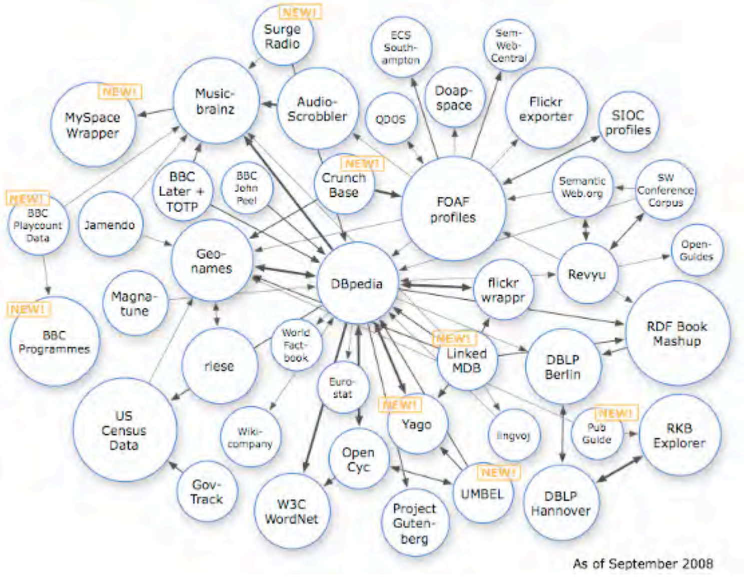


Figure 1. Linked data on the web (from [2])

In practice, this interlinking is not quite as easy as the figure suggests as some data is partial (e.g. in the DBpedia data for ‘Athens’ the *word* ‘Greece’ is mentioned, but is not linked to a semantic ‘Country’ entity), and while classes and relations are common through shared ontologies, the same entity (e.g. the City ‘Athens’), is typically represented by different URIs in different data sources, so some resolution and mapping is needed.

As well as these core Linked Data sources there is an even greater volume of data with public APIs, including data storage sites such as Freebase [29] and Google spreadsheets [30]. In some cases (e.g. Freebase) these have representations that either use ontologies, or have similar form, but do not use standard ontologies and thus are, in principle, harder to interlink with other data sources. It is likely that many of these will adopt Linked Data philosophy over time or that wrappers will be constructed by third parties, so that these offer a larger potential source of data.

Finally, while much of the web is designed to be human readable, this does not mean it cannot be accessed automatically. It is estimated that the vast majority of web-accessible information is ‘hidden’ in backend databases, but only available through bespoke web forms. This is variously called the invisible web, the hidden web, or the deep web [31,32]. Hopefully over time more of this will also become available either through the owners adding RDF interfaces, or through third party wrappers, or through automatic means [33,34]. Even for ‘ordinary web pages, it is suggested that around half the material is within some form of template [35] and text-mining techniques can enable semantic information to be extracted even from plain text [36]. Even where analysis of single pages is ambiguous or unclear analysis of large numbers of documents may yield more reliable information, as with Google Sets [37].

It is not yet clear whether the future of the web will be a pure semantic web approach of URI-linked data or one of more diverse data sources linked through wrappers and mappings. However, either way, for the purposes of this paper, we will assume that the data available acts like pure linked data. It may be that some of the entity linkages are inferred through mappings and rules, but if so we assume that this has happened prior to loading into a local graph.

2.5. *Web-Scale Reasoning*

There are a number of proposals for large-scale web reasoning including work emerging from the EU Large Knowledge Collider project (LarKC) [38]. Fensel, van Harmelen and the LarKC team propose a sampling based approach [3,39], which like our own work leads to defeasible reasoning using partial information. As in our approach they assume that bounded rationality [40] is essential when reasoning over very large knowledge bases.

```
do
    draw a sample,
    do the reasoning on the sample;
    if you have more time,
         and/or if you don't
         trust the result,
    then draw a bigger sample,
repeat
```

Figure 2. Sampling-based reasoning (from [3])

The concept of bounded rationality was originally introduced by Simon to describe the way we as humans think about the world [40], neither waiting until we have all the relevant information not even fully deducing all the logical consequences of our knowledge, but instead acting on partial information and partial reasoning. The explanation for this is not laziness or ‘poor’ reasoning, but necessity and efficiency: gathering information and thinking about it are both expensive, taking time and effort, and are typically not worth the additional gains. Elements of bonded rationality are found in many algorithmic approaches developed since the 1970s including simulated annealing, neural networks and genetic algorithms, which all aim to produce ‘good enough’ results for reasonable costs. In contrast to these nature-inspired algorithms, LarKC effectively mixes traditional deterministic reasoning with statistical methods. This kind of analysis has proved successful in various fields including primality testing in cryptography [41] and model counting [42].

Anadiotis et al. [5] look at a peer-to-peer architecture proposing that queries are broken into portions and distributed to the servers maintaining the relevant data based on the ontologies used in data at the servers. Proposals for *semantic web pipes* [43] and *stream reasoning* [44] similarly envisage being distributed across servers. The *augment* API in Talis' semantic web application platform [45] operates in a similar fashion allowing queries generated in one triple store to be augmented with knowledge from another. These approaches have some similarity to Google MapReduce [46], which has successfully applied functional programming techniques to very large-scale data processing. MapReduce assumes homogeneous and replicated information; however other forms of web-scale reasoning including stream and pipe approaches assume heterogeneous stores where the problem is knowing which stores have the required information and potentially localising computation to the relevant stores. The issues of ontology authoritativeness and reasoning scalability are discussed in [47], where a rule-based forward-chaining reasoning scheme is adopted.

While many of the systems and proposals for web scale reasoning are focused on distributing the reasoning or computation, some, including our own approach, assume a single reasoning engine drawing in information as required. The OBII semantic web query answering system works in this manner [6,48]; it has a repository of meta information about data sources and ontology maps to deal with disparate ontologies, then draws in information from different sources as required for the query being processed.

Arguably the most successful form of web-scale reasoning is Google page rank [49], which is effectively using a form of sub-symbolic reasoning. In fact, the simple page rank algorithm operates in a very similar way. Page rank uses linear spreading of 'rank' between web pages leading to a single stable global pattern of rank. In contrast, spreading activation is attempting to create a pattern of activation dependent on the initial activation, and so uses non-linear functions to prevent 'capture' by the eigenvectors of the linear approach. However, despite these differences, the success of Google page rank certainly suggests that other forms of sub-symbolic reasoning have potential.

3. Spreading Activation over Personal Ontologies

Having at hand a personal ontology that captures the entities of interest to the particular user and the relationships among them, we can simulate the spreading activation procedure to identify the entities that can be of interest to the user in a particular context. The basic idea is as follows: when the user performs an activity, some entities in the personal ontology may be referenced in the context of this activity – e.g. when the user reads an e-mail, the *sender*, other *recipients* of the same e-mail, or a *project* whose name is cited in the e-mail body are such candidate entities. These entities are said to receive *immediate activation*; afterwards, through the relationships established in the ontology, part of this activation can be spread to other connected entities within the ontology. When the algorithm completes, entities that have received a sufficiently high activation (above a certain threshold or the *top-k* ones) can be considered as the most prominent candidates for the user to perform subsequent activities on (e.g. reply to the e-mail; open the correspondence file with another recipient; or go to the project's document repository). In this respect, spreading activation can be employed in the context of Task Information Management (TIM) to provide context inference to tools that support TIM. In the following paragraphs we will briefly discuss how spreading activation can be applied on personal ontologies.

3.1. *Timescales in Human Memory and User Interaction*

Although the mechanisms of human memory have not been fully decoded yet, a number of relevant theories have emerged that explain different aspects of its structure and operation. A prominent model has been proposed by According and Shiffrin [50], according to which there are two distinct memory stores: short-term memory (known also as *working memory*), and long-term memory. Short-term memory corresponds to the things we are currently thinking about, and expires after a brief time period (10-30 secs), while its capacity is also limited (5-9 *chunks*) [51]. Long-term memory, on the other hand, corresponds to things we have learnt and remain for an indefinite amount of time (possibly for ever). Its capacity appears to be almost limitless, and items in it are organized mainly in terms of semantics, accommodating however procedural knowledge and images. Recent studies have proposed an additional intermediate memory store, termed *long-term working memory* [52] or *mezzanine memory* [1,53], storing information regarding the current situation.

Similarly to these three levels of memory, we may identify three timescales regarding the user interaction with a system: first, we can consider the full set of items that are of interest to the user, and have been modelled in the user's personal ontology; these items roughly correspond to the human long-term memory. Second, we can consider the items involved in the *current activity* of the user (e.g. items in the e-mail currently being read), which roughly correspond to the *working memory*. And, finally, we can consider items involved in recent history of the user's activities, which roughly correspond to the long-term working memory; these may provide a broader context of the user's activities, e.g. if a user reads an e-mail regarding an upcoming project meeting in London and then visits an airline reservation site to book a flight to London, then both activities can be contextualized as part of a more generic activity related to the project (participating in a project meeting).

One additional thing that must be taken into account is that not all items are equally important to the user: for instance, when considering long-term memory, one's own address is more important than the address of the plumber; analogous differences in entity importance can be also observed for short-term and medium-term memory.

3.2. *Accommodating Spreading Activation Information in a Personal Ontology*

In order to reflect which entities are currently active in a certain memory/interaction level of the user and the perceived importance of each such entity, we should extend the personal ontology model to include this information. To this end, each entity within the personal ontology includes the following additional properties:

- **STA** (Short-Term Activation), indicating that an entity is currently active
- **MTA** (Medium-Term Activation), indicating that an entity has been recently active (and could also still be)
- **LTA** (Long-Term Activation) to things that are important to the user in the long term.

All the properties above acquire numeric values to indicate how important the particular entity is deemed in the respective memory/interaction level of the user, while the value of zero for a specific property indicates that the item is not present within the particular memory/interaction level. We will also use an additional "trigger activation" property, **IA** (Immediate Activation), corresponding to the things that are in some way important directly due to the current task/interaction; for example, the ontology entities (classes and instances) that are recognized in the currently viewed e-mail or web page. This property will facilitate the operation of the spreading activation algorithm, described in the next sub-section. In order to accommodate these properties (STA, MTA, LTA and IA – as well properties IN and MAXLTA which will be discussed later) in all ontology instances, we have extended the definition of the template class (STANDARD-CLASS Protégé [54]) to include these properties, and from there these properties are inherited to all ontology instances. All additional properties are of type *float*.

In order to simplify the presentation of the spreading activation algorithm, we will consider that the inverse of each relation is explicitly recorded in the ontology schema e.g. if the ontology includes entities "John" and "Mary" and these are connected with the (directed) relationship *father("John", "Mary")*, then the ontology also includes the directed relationship *daughter("Mary", "John")*. In the implementation of the algorithms, activation is spread in both directions through a relationship even when there is no defined inverse, but for the sake of exposition, we assume that both are there.

We will also consider that relationships bear a weight (or *strength*) LTW, which is directional, allowing different weights depending on which direction the relation is traversed. LTW is again accommodated in all relationships within the ontology, by extending the respective template class in Protégé, namely STANDARD-SLOT.

The newly introduced properties listed above (STA, MTA, LTA, IA, IN, MAXLTA and LTW) describe the spreading activation-related aspects of the ontology elements, constituting effectively *meta-information* for these elements.

3.3. *Spreading activation algorithm*

The spreading activation algorithm operates on the personal ontology, as enhanced with the properties STA, MTA, LTA, IA and LTW listed in sub-section 3.2, and includes rules for updating the activation levels of the entities within the ontology. Updating includes passing activation between shorter-term and longer-term memories and modelling decay of memories, in the absence of triggering activations. In the algorithm description and discussion presented below, we will use the following notations:

- *IA(e), STA(e), MTA(e), LTA(e)* are the instant, short-term, medium-term and long-term activation levels of a particular entity *e*.
- For a particular relationship *r*, we will denote as *LTW(r)* the weight of the relationship (i.e. its perceived importance). We will also denote as *LTW'(r)* the value of LTW(r) divided by the number of entities to which *r* points to (the *fan-out* factor of *r*). For example, if *r* is the relationship "member state" between the entity "European Union" and the entities corresponding to countries, *LTW'(r) = LTW(R)/27*, since the relationship connects entity EU to 27 other entities.

The basic steps of the spreading activation algorithm (summarized in Figure 3) are as follows:

First weights are computed for relationships determined largely by fan-in/fan-out and also those entities with initial activation (`IA(e) > 0`) are added to an '`Active Set`'.

Then a number of iterations are performed calculating the short term activation of each entity (`STA(e)`) based on spreading from the '`Active Set`'. The precise formulae used for this are described in section 3.3.1

At each iteration, any entities with sufficiently high activation are added to the '`Active Set`'.

The termination condition for this process is discussed in section 3.3.2.

Finally, if the activation of any entities is sufficiently high the long-term and medium-term activation (`MTA and LTA`) are updated.

```
1. Initialize appropriate weights and activations
2. Create a set with the currently active entities (entities
    e with IA(e)> 0), Active Set
3. Repeat
    Compute STA(e) for the entities in the Active Set as
    well as their related ones
    For the related entities whose STA exceeds a threshold,
    add them to the Active Set
   Until <condition>
4. Update MTA and LTA activation weights if appropriate
```

Figure 3. Basic Outline of the Spreading Activation Algorithm (from [1])

3.3.1. *Updating short-term activation*

The short-term activation for a specific entity stems mainly from the following two factors: the first is the direct appearance of the entity in the current task/interaction (e.g. its presence in the e-mail just read), corresponding to IA(e). The second factor is the entity's relationship to other entities that are currently in the short-term memory, e.g. when a scientist considers a paper he has authored (and thus the entity corresponding to the paper has a high STA), the entities corresponding to the paper co-authors or the forum the paper has been published in become active. The second factor will be termed *incoming activation* and will be denoted as IN(e); we compute it through the formula

$$\text{IN(e)} = \sum [\text{LTW'(r)} \times \text{STA(e')}],$$

where the sum is over every entity e' connected to e via a relation r in the ontology

This effectively states that the incoming activation for an entity *e* is derived from the entities *e'* that are related to it, and are currently in the short-term memory. Each such entity *e'* contributes to IN(e) proportionally to the strength of the relationship between *e* and *e'*.

Besides IA(e) and IN(e), the computation of STA(e) should take into account the importance of *e* in the current task context [corresponding to MTA(e)] and the overall importance of *e* [i.e. LTA(e)]. Combining all the above,

$$\text{STA(e)} = \text{S(f (IA(e), IN(e), MTA(e), LTA(e)))}$$

Function *f* must count IA(e) strongly, since entities directly referenced in the current task are the most active ones in short-term memory. Moreover, MTA(e) and LTA(e) should be taken into account only if either IA(e) or IN(e) is non-zero. This last requirement is to ensure that the eventual activation is determined by the initial activation. If MTA and LTA were too strong they could swamp the effects of the initial activation leading to a stable, but undifferentiating activation.

Thus, one of the simplest plausible choices for f would be:

$$\text{f(ia, in, mta, lta)} = (\text{A} \times \text{ia} + \text{B} \times \text{in}) * (1 + (\text{C} \times \text{mta} + \text{D} \times \text{lta}))$$

The result of function *f* is passed through a sigmoid function [55]:

$$S(sta) = \frac{1 - e^{-sta}}{1 + e^{-sta}}$$

The sigmoid serves to emphasises the difference between large and small activations and caps the largest. The equation for STA is recursive and is applied on the set of activated entities of each step.

3.3.2. *Terminating spreading of activation*

Since spreading activation is by nature recursive, a *termination condition* must be established to break the recursive step. The two most prominent options are:

(a) to apply the recursive step until the ontology reaches a stable state. Note that since the ontology contains loops (recall that for any relation its inverse also exists, forming thus a loop of length two; additional loops will also exist in the ontology), we cannot expect that at some step *all* activation transfers will be zero. Thus, we consider a state as stable when *all* activation transfers in some step of the recursion fall below a certain threshold th_{stable}, which can be defined either as an absolute value (e.g. $\delta(STA(e)) < 10^{-4}$, where $\delta(STA(e))$ denotes the increment of STA(e) in a particular step of the recursion) or as a ratio of the computed increment divided by the current value of the receiving entity's STA (e.g. $(\delta(STA(e))/STA(e)) < 10^{-3}$).

(b) to apply the recursive step for a specific number of iterations (e.g. 20).

In [1], constrained spreading activation (option b) has been followed, as also suggested in [56]. In section 5 we will discuss reasons to suggest that the ontology graph is a 'small world'. That is the distance between any two entities is likely to be small, where 'distance' as measured by the number of relationships traversed to get between them. If this is the case, then only a relatively small number of iterations are needed to ensure that activation could spread right across the graph. In experiments reported in section 5 on large ontologies (millions of triples), we observed informally that there was little change in activation levels after 10-20 iterations.

While the termination condition is about how *long* to continue with the spreading activation, later in this paper (section 4), we will discuss how thresholds can be used to limit how *far* the activation spreads through an ontology.

3.3.3. *Updating MTA and LTA*

In the algorithm presented in section 3.3, after the loop that computes and updates STA, MTA and LTA are updated. MTA is incremented if STA exceeds a certain threshold:

$$\text{if } (STA(e) > threshold_{STA})\ MTA'(e) = MTA(e) + \delta_{MTA}$$

and similarly for LTA:

$$\text{if } (MTA(e) > threshold_{MTA})\ LTA'(e) = LTA(e) + \delta_{LTA}$$

For a complete discussion on the how the thresholds of STA and MTA are set, as well as how the values of δ_{MTA} and δ_{LTA} are derived, the interested reader is referred to [1]. While the provisions above cater for incrementing the values of MTA and LTA, we must also include provisions for their decay, i.e. their value should be decremented when the entities are not active for a period of time. The mechanisms for their decay are considered differently, due to the different nature of medium-term and long-term memory (the human capacity for dealing with different subjects in a period of time is limited, as opposed to the almost unlimited capacity of long-term memory).

To model the limited capacity of medium-term memory, we define a constant MaxMTATotal to represent the maximum value for the sum of all MTA weights in the ontology, and the following process is performed every *T* steps:

```
1. The total amount of MTA increase over the T steps, $s_{MTA}$,
   is recorded
2. We set $\lambda_{MTA} = s_{MTA}$ / MaxMTATotal as the decay factor
3. For every entity e, the new MTA is computed:
   MTA'(e) = (1 – $\lambda_{MTA}$) * MTA(e)
```

Figure 4. Process to decay MTA, performed every *T* steps

The number of steps *T* after which the decay process should be performed, as well as the value of MaxMTATotal should be set after taking into account the needs of the application at hand.

Regarding the decay of LTA, we should consider that LTA reflects the long-term importance of entities, it should be ascertained that the decay does not result in important things having their LTA value gradually returning to zero. This can be achieved by introducing a rule that the LTA of an entity never decays to less than a percentage (n%) of its maximum value. Thus, we denote as maxLTA(e) the maximum LTA value an entity e has ever received. Additionally, we introduce two constants, λ_{LTA} as the decay constant that depends on the time interval between each decay and minPerc as the minimum percentage of the entity maxLTA value that the LTA of an entity may reach when decayed. The LTA decay is computed using the following process:

```
At the designated time points, for every entity e:
    if (LTA(e) > maxLTA(e)) {maxLTA(e) = LTA(e)}
    minLTA_e = minPerc * maxLTA(e);
    deltaLTA_e = $\lambda_{LTA}$ * (LTA(e) - minLTA_e)
    newLTA_e = LTA(e) - deltaLTA_e
    if (newLTA_e >= minLTA_e) LTA(e) = newLTA_e
    else LTA(e) = minLTA_e
```

Figure 5. Process to decay LTA

Note that the amount that LTA is decremented by (deltaLTA_e) is proportional to the difference between the current value of LTA(e) and the minimum allowed value for LTA(e), thus LTA dropping rate is smaller when the current value approaches the minimum value and higher when the value of LTA has been significantly incremented in the recent past (and has not been refreshed).

3.3.4. *Dealing with Relation Weights*

Relation weights are a very important issue in the spreading activation framework, since they play a dominant role in computing the entities' incoming activation IN(e). We can consider three levels of relation weights, which play a part in regulating the spreading of activation between entities:

1. The relation as a whole, which is expressed by the relation's Long-Term Weight – LTW (e.g. the "friend" relation will have a higher LTW than the "acquaintance" relation).
2. Weights on a particular instance of a relation, that is for a specific e1, e2 with a relation r between them, we could assign a weight dependent on:
    - An *a-priori* choice of the user – e.g. if there is a "friend" relationship, the user could assign higher weights to "better" friends.
    - Whether the relation was important in spreading activation
    - Whether both e1 and e2 have received high activation during some period.
3. Weights on the relation for an individual entity. We can quantify this through LTW'(r), defined in section 3.3, which arranges for "splitting" the spreading of the activation through a particular relation to all entities it connects. This is a more coarse-grain option for computing the weight of particular relation instances, as opposed to option (2) which considers individual instances separately.

Similarly to activation levels, relation weights can also be adjusted; these adjustments will reflect the observations on how often entities connected through the relationship become active together. An approach for updating LTW of relations is presented in [1].

3.3.5. *Effectiveness of Spreading Activation*

A preliminary evaluation has been conducted on the spreading activation algorithm as described above, to verify its effectiveness. The preliminary evaluation included 37 tasks, and within each task specific entities were stimulated through immediate activation. Then, users were asked to classify the entities proposed (i.e. received an STA value of 20 or greater) by the spreading activation algorithm into one of the following categories (a) relevant and useful, (b) relevant but not useful and (c) irrelevant. Users were also asked to designate whether some ontology entities were *important* in the context of the current task and were not proposed by the spreading activation algorithm. The results of this preliminary evaluation are as follows, while for more details on the experiment, the interested reader is referred to [1] and [57]:

- 59% of the proposed entities were characterized as *relevant and useful*.
- 33.3% of the proposed entities were characterized as *relevant but not useful*.
- 6.1% of the proposed entities were characterized *irrelevant*.
- In 14 of the sub-tasks, 1 entity identified by the user as *important* was not proposed, whereas in 4 sub-tasks 2 important entities were not proposed. In the remaining 19 sub-tasks *all* important entities were proposed.

Measuring the effectiveness in terms of the standard information retrieval metrics, namely *precision* and *recall* [58], *recall* ranges from 78% (two sub-tasks) to 100% (19 subtasks) with an average of 94%. The minimum *precision* value encountered was 68% (one subtask), while in two other tasks the obtained value was 78%; in other subtasks precision values ranged from 82% (two subtasks) to 100% (19 subtasks), with an overall average of 92%. Finally, the f-measure (a combined metric involving both precision and recall) ranged from 75% (one subtask) to 100% (11 subtasks), with an average of 93%. Since, however, the proposed approach was evaluated in this experiment not as a generic information retrieval support infrastructure but rather as an underpinning for assisting user tasks, we should probably calculate the *precision* and *recall* metrics by considering as “relevant documents” the results that are *both* relevant *and* useful, since these are the ones that are bound to assist the user task at hand. Under this approach, *precision* ranges from 40% to 78% with an average of 59%, while *recall* ranges from 67% to 100%, with an average of 91%. Finally, the *f-value* has a minimum of 53% and a maximum of 88%, with a mean value equal to 71%.

Results are thus promising, however a more thorough evaluation and an elaborate parameter tuning are underway. Performance-wise, the STA computation step as well as the MTA and LTA update steps were performed in less than 3msec on an ontology containing 75 classes and 214 instances; therefore -at this ontology size– it is feasible to perform the STA computation and MTA and LTA update steps almost after every user activity and propose to the user prominent entities and/or activities.

4. Web-Scale Spreading Activation

Spreading activation can, in principle, involve work over the entire ontology, and certainly any part of it. For small ontologies this is not a problem but for larger ontologies the cost is, in worst case, proportional to NxDxR where N is the number of entities in the ontology, D is the average degree of connectivity for an entity (number of relation instances involving the entity) and R is the number of spreading activation iterations.

For the personal ontologies we have been considering so far, this is not a problem as all the information has been explicitly entered by the owner of the ontology and is thus relatively small. Indeed optimizations have not proved necessary, as simple sequential passes have been fast enough. The hand-crafted part of the ontology will grow over time, but probably slowly enough that it can always be dealt with in-memory and with relatively straightforward algorithms. However, this hand-crafted part of the personal ontology is just the core linking to further personal resources on the user’s desktop (files, emails) and in the user’s web-based services (Flickr, del.icio.us), and in addition to external resources for workgroup or corporate information, and ultimately to the whole web.

For the former, personal sources of information, it is reasonable to assume that a complete meta-information may be gathered into some repository on the user’s own machine, as is done in various semantic desktop projects [10,11,59]. However, even then it is likely that the size of the ontology will be greater than can fit in main memory. More critically, as we consider shared information both corporate and full web, we have to assume that the majority of the information is not only external to the user’s personal machine, but is so large that it could never be. N is effectively unbounded.

We will look first at the simpler case when the ontology is large and in-memory and then use this to consider the more complex case where we wish to use spreading activation for ontologies, such as the web, where the complete ontology is too large

4.1. *Limiting spread in large memory-resident ontologies*

If N is very large and the ontology is cliquey then the costs of spreading activation may not be as large as NxDxR and instead be N_rxDxR, where N_r is the number of entities reachable in *r* steps from the original activated entities. Note, at each step N_r is the maximum number of entities that can have non-zero activation as the rest will not have been 'touched' yet by the spreading. However, most ontologies will be 'small worlds' and so N_r will be close to N for relatively small r. So, we need to artificially introduce limits.

Threshold-based limit – A threshold can be imposed on spreading steps; that is only spread outward if the activation at an entity exceeds a certain threshold *t*. This will have a significant effect as the time becomes bounded by $N_r(t)$xDxR where $N_r(t)$ is the number of entities with activation exceeding threshold *t* after *r* iterations. With a small, but non-zero threshold it is likely that $N_r(t)$ is significantly smaller than N and, in particular, will only scale slowly as the ontology gets larger (we will examine this assumption further in section 5.) Note that this requires keeping track of all activated entities if we are to avoid linear searches of all entities. However, the linear scan may turn out to be faster until N is very large. (The linear scan would be O(N) whereas some form of activated nodes list would be $O(N_r)$, but the time per iteration for the latter would involve something like creating a linked list, whereas the former would be simply scanning for entities with high activation.)

Cap-based limit – A variant on this would be to choose a fixed n and only spread from the n most activated nodes. This has the advantage of establishing a cap on time per iteration, but does mean keeping a list of entities part-sorted by activation, that is, at worst, an extra O(n x D + n x log(n x D)) cost per step. If $n < N_r / \log N_r$ this will still be cheaper, but anyway the sorting does not have to be perfect, so actual cost is likely to be smaller.

Note that adding a threshold or cap changes the semantics of spreading, the results will be similar but not identical to spreading without a threshold. We will return to this issue empirically in section 5. However, it is worth noting that many neural models include some form of threshold for signal propagation. We have not needed to do this in our spreading activation as the sigmoid function basically 'squashes' low activation and makes some effectively zero. However, if anything, a threshold is more similar to the way our own brains work.

The choice between threshold or cap is likely to be pragmatic. Certainly during our own experiments (described in section 5), we found that outputs were very stable for different threshold levels, and thus suggest that cap-based limits (effectively a variable threshold) are thus unlikely to behave differently.

4.2. *Non-memory resident ontologies*

As noted, even information extracted from personal resources such as email archives may become too large to fit within main memory and clearly the web is too large! So we have to consider strategies for dealing with much larger ontologies.

Measurements of the web [60] suggest that about 75% of pages are connected (linked to or from) a single strongly completely connected component (the SCC) comprising about ¼ of the web. For pages in or connected to this SCC, the average distance in terms of undirected links between pages is 7 links. While the web of data is not sufficiently developed to be able to predict the equivalent figure for it, it is reasonable to assume that it too will comprise a relatively small world. It may contain some disconnected components, but, if the linked data vision becomes reality, the majority of entities will be linked to the whole. We can therefore assume that WN_r, the number of entities in the web of data at distance *r* or less from an initial activation entities, is likely to be very large if not comprising the majority of the web.

For generic global calculations such as Google PageRank for web pages, it is acceptable to effectively reason over the *whole* web, but for more bespoke queries, and especially our own application area where we want fast per-user-interaction update of context, we need to be reasoning over just the *relevant* web.

By its nature spreading activation tends to have a non-local effect, which is likely to interact poorly with non-local access, at worst touching every entity and triple in the ontology. Some means are therefore required to limit the impact and spread of activation in order to avoid this large scale flooding of the ontology.

Happily, the number of high activation entities is substantially smaller than the total ontology and so limiting the number of activated entities, whether using threshold or cap, has the potential to help significantly as only the activated entities need to be brought into main memory.

For context inference this is particularly appropriate as the active entities are also be expected to change only slowly over time. Furthermore, if we are using STA/MTA/LTA scheme, then it is likely that MTA as well as STA can be maintained entirely within main memory, with only LTA recorded on disk. However, LTA can be stored on local disk, even if the entities it refers to are distant (see also section 4.5 for loading rules for LTA).

In fact things are slightly more complicated as we can only know the activation of an entity if it is in memory to participate in memory resident spreading activation. That is the choice of whether to bring in a new entity can only be made based on the entities and relations *already in main memory*. We therefore need some form of *fetch rules* to determine what to bring into memory and also *discard rules* to decide what and when to purge data to make room for new.

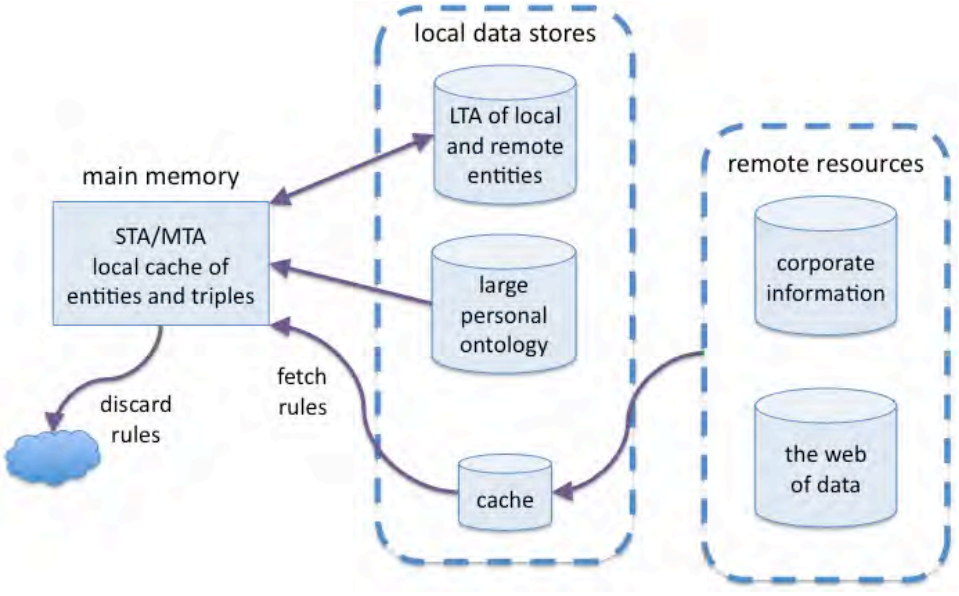


Figure 6. Proposed architecture

Figure 6 shows the main components needed for this. This shows the personal ontology and also various remote resources. In many ways the personal ontology and remote resources can be treated uniformly, but there are some differences as discussed in section 4.8. A disk cache is also shown for remote resources, but as this is a standard feature we do not consider it further. There is also a local persistent store for LTA. Note that this may include LTA for entities in remote stores as well as those in the personal ontology. In this figure and the rest of this section, relation instances are considered to be expressed as *triples* <e, r, e'>, denoting that entity *e* is connected to entity *e'* via a relation *r*, in keeping with semantic web usage.

We have not included a symbolic reasoning engine explicitly in Figure 6, as our focus is on the contextual reasoning of the spreading activation. However, in section 6.1 and 6.2 we discuss how symbolic reasoning might be integrated into this picture. Of course the remote resources may themselves have some level of reasoning support; in this case we effectively treat primary and inferred data uniformly. If the remote reasoning itself involves some level of fuzziness or uncertainty and this is passed on as provenance, this could be used to modify weights within the spreading activation, but we will not consider this in detail here.

4.3. *Entities 'in memory'*

Actually the idea of whether an entity is 'in memory' is itself slightly problematic, while it is the entities that are activated in the spreading activation, in a pure ontology-based system an entity is no more than its identity and the triples/relation-instances involving the entity. Strictly the question is what triples are in memory, what entities are mentioned in some triple in memory, and what proportion of the triples mentioning an entity are in memory

If we let T_o be the set of all triples in the ontology (both the personal ontology and non-local resources including the whole web) and T_m are the triples currently in memory, we have:

$$T_m \subseteq T_o \subseteq E \times R \times E$$

Where E is the set of all entity labels and R the set of all relations. We can then define:

E_o = entities (T_o) – the entities present in the full ontology
E_m = entities (T_m) – the entities mentioned in the triples in memory

where:

$\text{subjects}(T) = \{ e \mid \exists <e,r,e'> \in T \}$
$\text{objects}(T) = \{ e \mid \exists <e',r,e> \in T \}$
$\text{entities}(T) = \text{subjects}(T) \cup \text{objects}(T)$

For an entity *e* mentioned in main memory, the triples in T_m referencing *e* may be a more or less complete subset of the triples in T_o referencing *e*. At one extreme T_m may contain only one of these triples from T_o, while at the other it may contain *all* the triples from T_o that include the entity. If the latter is true we can say the entity *e* is *complete* in the ontology:

$$\text{complete}(e) = \forall <e_1,r,e_2> \in T_o \;:\; e_1=e \vee e_2=e \;\Rightarrow\; <e_1,r,e_2> \in T_m$$

More generally we might be interested in a particular subset of entities E' and relations R' and whether a particular entity e that is mentioned has all triples relating it to entitles in E' through relations in R'

$$\begin{aligned}&\text{complete\_wrt}(E',R')(e) = \\ &\quad \forall <e_1,r,e_2> \in T_o \;: \\ &\qquad r \in R' \wedge ((e_1=e \wedge e_2 \in E') \vee (e_1 \in E' \wedge e_2=e)) \Rightarrow <e_1,r,e_2> \in T_m\end{aligned}$$

As shorthand we shall use complete_wrt(E') to mean complete_wrt(E',R) where R is the set if all possible relations.

One kind of ‘being in memory’ for an entity is to ask that it is *complete* in the sense above of having every related triple. This is equivalent to performing an RDF DESCRIBE Query from the disk triple store [61]. Alternatively we may simply include all the links between it and things in memory (that is complete_wrt(Em)).

4.4. *Fetch rules: choosing which triples to load into memory*

We can follow the general principle of bringing in data related to highly activated entities, but there are a number of variations, several or all of which can be applied.

Filling-in rule – If an entity's activation exceeds a critical threshold t_{fill}, we retrieve all triples linking that entity to others in main memory - this is we make the entity complete_wrt(E_m).

Ripple-out rule – If an entity's activation exceeds a threshold t_{ripple}, we retrieve *all triples* that include it.

If we apply both of these fetch rules then we need $t_{ripple} > t_{fill}$.

These rules are both focused on adding information about the entity under scrutiny. The ripple-out rule is potentially problematic if the entity is highly connected; for example, if it were a country, say 'Greece' then we would end up drawing in every person whose country of birth is Greece, as well as the city Athens as the capital of Greece, Greek as its language, etc.

A more conservative rule than ripple out would be to preferentially include triples with smaller fan-out (capital, language rather than 'place of birth of'). This is similar to what we do for spreading activation itself, and so we can think of rules that are a form of 'look ahead; retrieving triples that would get a certain level of activation 'if they were there'.

Look-ahead rule – For any entity *e*, take all relations *r* that may have *e* as subject or object (based on typing), but for which we do not yet have all instances in main memory. Assume we know the fan-out $f_{r;e}$ for relation *r* from entity *e* (either use average for the relation, or more specific count if it is known). Use $f_{r;e}$ and the current activation of *e* to calculate what activation *a* would be spread to entities connected via relation *r* if they were present in memory. If *a* exceeds some threshold t_{look}, we retrieve *all triples* <e,r,?> or <?,r,e> (depending on whether e is subject or object of relation). Note for relations such as 'child-of' that operate between entities of the same type, we must apply this rule differently depending on the direction as the fan-out is different.

In section 5 we will see that adding the look-ahead rule to the ripple-out rule can significantly reduce the number of nodes fetched from remote resources whilst making little discernable difference to the results.

In all cases we may bring in additional triples if, for example, they reside on the same disk block as the target triples and then decide which to keep using for this decision a lower threshold than when deciding what to bring in. This is unlikely to apply if we are operating through a high-level interface to remote storage (e.g. SPARQL), but may be a potential optimisation if we have lower-level access to a local ontology store.

4.5. *Choosing when to load LTA into memory*

In order to perform spreading activation we need to have LTA and MTA for a node and also to store STA. So, at some point, we need to create some form of record for the entity that includes retrieving its LTA from disk.

The *filling-in* rule doesn't add new entities, so that is not an issue, but both the *ripple-out* and *look-ahead* rule have the potential to mention fresh entities that we were previously not mentioned by triples in main memory. For these new entities we need to decide whether to in addition retrieve relevant meta-information (in particular LTA) from disk.

When we have applied the *look-ahead* rule, we are expecting that any new entities mentioned in the retrieved triples may have sufficient activation to immediately be part of the SA, so it is reasonable to retrieve their LTA at the same time.

In the case of *ripple-out* we may be retrieving many entities that potentially may never have high activation. In such cases it may be worth creating a stub internal record for the entity to record STA, but then only if STA exceeds some threshold t_{meta} should we bother to retrieve the full meta-information (e.g. LTA, cached fan-in/fan-out counts)

4.6. *Discard rules: deciding what to purge from main memory*

We can apply a similar rule to decide what to remove from main memory. If the activation of an entity *e* (STA and maybe also MTA) is below some threshold(s), then we purge *e*. By purge this means removing triples that include *e*, except those that would be immediately be brought back in by one of the fetch rules above. As we remove the entity and triples, we need to make sure that any LTA information is preserved (and MTA if we have removed based on STA activation only).

```
PseudoCode
    foreach entity e such that $STA(e) < t_{purge}$ and $MTA(e) < t_{MTApurge}$
        if LTA(e) has changed update LTA(e) on disk
        if MTA is persistent and MTA(e) has changed update MTA(e) on disk
        foreach triple t where e is subject or object of t
            if filling-in or ripple-out or look-ahead rule applies to triple t
                leave t in main memory
            else
                delete t from main memory
        if no triples remain mentioning e
            remove meta-information record for e from main memory
        otherwise
            leave meta-information (maybe as stub)
```

In principle we would need some form of discard rule for the records of LTA of remote entities in persistent storage, however, the size of this may be small enough for it this never to be necessary. If the LTA is subject to periodic decay, then this will require a serial pass over the LTA store and so, if required, this would be an obvious time to purge LTA records for remote entities if the LTA dropped below some threshold.

4.7. *Optimising fetch and discard rules*

Longstanding Operating Systems memory management techniques can be applied either directly or in modified form to increase the efficiency of the fetch/discard techniques [62].

(1) When we need to replace an entity residing in the memory with another AND the "victim" entity has been changed (here principally the LTA of an entity), its disk image should be updated. However, this takes additional time and speeds down the process. To avoid having to write entities to the disk at the instant we need the memory space, a separate thread can arrange so that disk images of changed entities are updated, thus entities can be victimized without any delay at that point. This also improves system stability in cases of software/hardware crashes (updates are not lost),

(2) Operating systems use "high/low watermarks" for the rates of page faults: if we are getting too many page faults that we have not allocated enough pages to some particular process, while if we are having too few, then we have over-allocated pages. The analogy here is with "entity misses" i.e. attempts to process entities that are not in memory. If we are getting too many of those we could try to increase memory allocated to the spreading activation activity AND/OR we could set more strict thresholds to have less entities activated. The latter possibly leads to sub-optimal results, but this approach may be preferable to having optimal results computed in excessive time, especially when these results should be presented in the user interface.

4.8. *Special issues for the whole web*

While we have treated all sources equally in most of the above discussion, there are some differences between resources for which there is a level of relatively local control (both personal ontology and corporate data) and the web, which is unregulated and decentralised.

The basic concepts are the same. If George gets a mail from Vivi and the entity Person('Vivi') is sufficiently activated, then this might trigger the fetching of the triple lives_in(Person('Vivi'),City('Athens')) from the personal ontology and subsequent activation of City('Athens'). Similarly, if 'Athens' is sufficiently active we might draw in additional information about Athens from Geo-names or DBpedia.

There are also differences. Web data is spread over multiple sources and we cannot simply send queries to all such sources "do you know about entity *e*". However, if we have meta-information about the data sources (in particular about the classes of entities and relations that they describe) then we can send directed queries to a small number of relevant sources. An assumption of some form of meta-information is common to most proposals for web-based reasoning, this may be hand crafted [5,48] or obtained as the result of some form of web crawl [63].

Given accessing remote resources is more expensive than local resources the thresholds for fetching data should be higher (maybe related also to the source if some are slower or more expensive than others). Similarly the threshold for purging cached remote entities (from secondary memory) should be quite low, to avoid re-fetching remote data and certainly linked to the LTA so that entities with high LTA are more readily available. Note too that the presence of a cache means that remote access may include 'fetch if cached' as the threshold for obtaining locally cached copies can be lower than that to obtain truly remote data.

5. Testing Assumptions

Our methods for scaling spreading activation to the web rely on the following assumptions:

(i) the total 'working set' of entities and triples needed during SA is relatively small;
(ii) the number of external sources that need to be accessed is also relatively small;
(iii) restricting the SA by thresholding and other techniques described in section 4 does not significantly change the semantics.

To some extent similar assumptions to (i) and (ii) underlie most approaches to web-based reasoning with the exception of global algorithms used in search engines. For example, in Anadiotis et al. architecture for peer-to-peer reasoning over the semantic web [4], it is important that request are not sent to too many peers; similarly in OBII [48], which has a centralised approach drawing in information as needed, it is important that not too many sources are consulted. In section 5.1 we will look at some of the evidence from external sources for assumptions (i) and (ii).

In sections 5.2 and 5.3 we will consider evidence based on our own experiments for (i) and (iii), looking first at the behaviour of a small personal ontology in Protégé and then of a large linked-data RDF store.

5.1. *Evidence from external sources*

As justification for their approach, Anadiotis et al. [4] argue that there is a significant degree of locality in semantic web. They use data from Swoogle [64], the semantic web search engine, that suggests that most namespaces are only used in a very small number of documents (see figure 7), implying that most inference will likewise be limited.

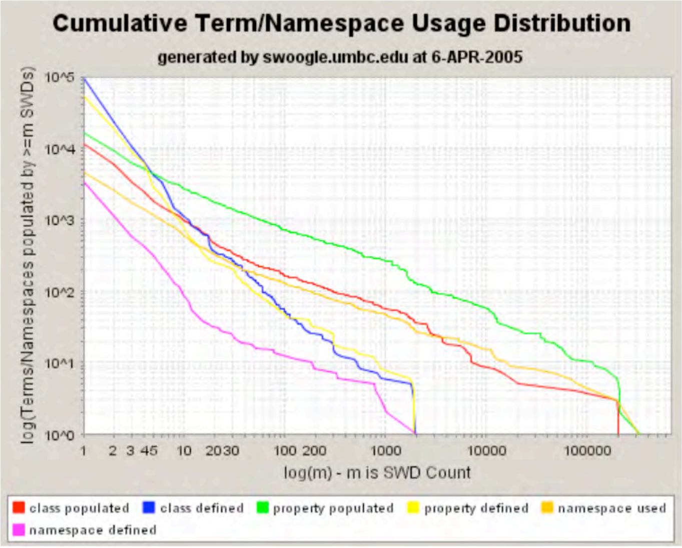


Figure 7. Cumulative Term/Namespace Usages Distribution (from [64])

However, there are a small number of namespaces with very wide usage. Some of these are obvious meta-description namespaces such as RDF, RSS and MVCB (MetaVocab vocabulary). However, others, in particular FOAF and other namespaces referring to personal details, tend to be used widely both in blogging web sites and also individual web pages: Again using Swoogle, Ding et al. [65] quote figures of over a million documents using FOAF in 2004. For these few but highly used namespaces it is likely that semantic search/aggregation servers such as Swoogle would be used rather than accessing the original data sources.

Further strengthening assumption (ii), Hogan et al. [47] suggest using only sources that are pre-designated as *authoritative* are taken into account to perform web-scale reasoning. “Topic distillation” [66] is a more flexible means for achieving the same goal; according to this approach we may query the web for all ontologies that can offer information on the topics of interest, but then process only results coming from the most “authoritative” sources; the metric of *authoritativeness* in this case is the number of “incoming links” to the ontology (effectively the number of ontologies that *include* this ontology to extend its terms or perform reasoning on it).

Several approaches attempt to take large ontologies and extract smaller modules form them. Cuenca Grau et al. [42] propose definitions of an ontology module based on concepts of conservative extension used in earlier formal literature. As well as being well founded formally, they show that their approach can lead to smaller modules compared to alternative techniques. Other techniques are based more on measures of sub-graph connectivity and so may be closer to the locality properties of SA, but they too report strong locality [67].

These general locality arguments underlying web-based reasoning also support the belief that as spreading activation attempts to draw in relevant entities, it will only need to hit a small number of servers.

However, spreading activation has particular properties that are different from query answering, in particular as it spreads potentially across *any* relation, there are no natural limits set by the schema of the initial activation. There are some similarities in symbolic approaches, for example, if a query includes higher order elements or if it refers to a class with many subclasses. However, it could be that symbolic queries are 'better behaved' in terms of locality than sub-symbolic approaches.

In some ways the opposite is true, thresholds can be adjusted dynamically to throttle excessive querying of remote sources, or cost-based approaches can be used to choose the best sources to include at any point. So in terms of practice, we can effectively enforce a level of locality. This is very similar to the sampling approach of LarKC [38], where they can choose a sample size and effectively limit the resources to those available.

However, while this means that computation cannot run wild, we are then left with the question as to whether this compromises the effectiveness of the spreading activation – does it give the right answer?

5.2. *Locality properties on a personal ontology*

In previous work [1], as described in section 3.5, we have assessed the *accuracy* of our spreading activation algorithm in terms of precision and recall, by comparing it with human selections of relevant entities. That is, we have some confidence that spreading activation works appropriately for ontologies without any external resource limits. We are therefore left with an issue of *robustness*: will the results of spreading activation change significantly when we introduce threshold-limiting spreading as described in section 4.

To answer this we have performed a number of experiments over the populated personal ontology using the algorithm described in section 3. For the purposes of these experiments LTA and MTA are set to zero as these will tend to stabilise STA activations and so zero values are the harshest test. Figure 8 shows the activation profile where a single entity has been given an initial activation. The graph shows the entities ordered by decreasing activation. The maximum activation level is 100 due to the non-linear sigmoid function used to limit high activations to 0-100 range. Examining the graph, there is a rapid drop off in activation, only a small number have activation more than 10 and the majority of entities have activation less than 1. Figure 9 shows a similar graph with two initially activated entities. Approximately twice as many entities are highly activated as for a single initial node as one would expect, but again there is a rapid fall-off in activation after a few most highly activated entities.

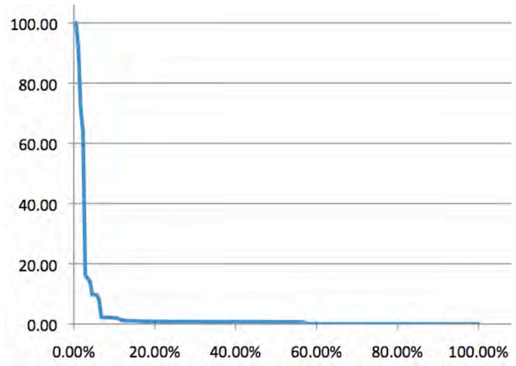


Figure 8. Activation profile – single activated node

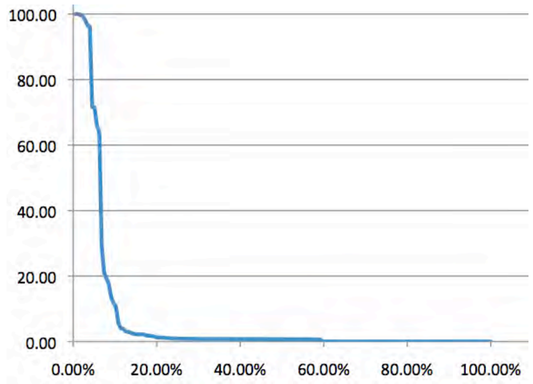


Figure 9. Activation profile – two activated node

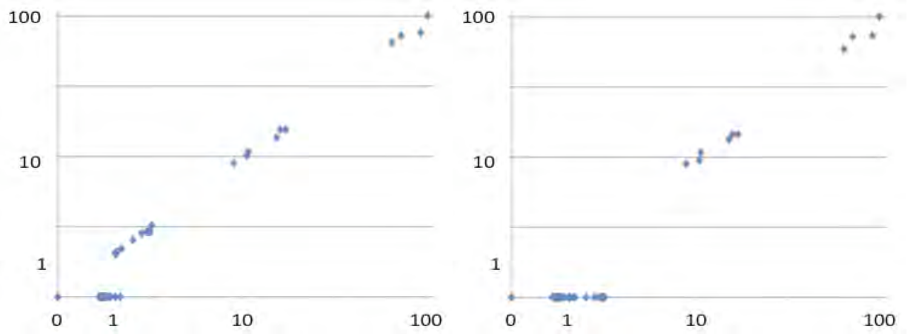
Of course the exact activation profile will depend on the structure of an individual's personal ontology, the entities given initial activation and any medium or long-term activation. However, the graphs are typical over the test ontology and we have seen similar results on preliminary studies using larger simulated ontologies. While not conclusive without larger-scale studies, these results do suggest, that, as expected, the spreading activation does have a relatively small working set of highly activated entities (assumption (i)).

Even if the working set is small, it may be that the pattern of activation in the working set depends critically on large numbers of entities with small activation. The spreading activation algorithm has been tuned (using fan-out weighting and non-linear sigmoid) to prevent strong feedback effects. The reason for this is to prevent 'greedy entities' that are always activated independent of the initial activation; that is in order to ensure *correct* behaviour. However, as a side effect, this is also expected to reduce the sensitivity of the activation pattern to large numbers of low activation entities; that is to help ensure *robust* behaviour.

In order to verify this, a simple threshold was introduced into the algorithm. If an entity has activation below this threshold it does not spread any activation to related entities; these correspond roughly to entities that might never have been ‘brought into memory’. Otherwise, exactly the same activation combination function 'f' and sigmoid function 'S' are used as in section 3. Figures 10 and 11 show the impact of this on the levels of activation with different threshold levels and different numbers of initially activated entities. Given the 0-100 range and the patterns of activation evident in Figures 8 and 9, values for a threshold of around 1% would appear reasonable. However, to test the robustness of the algorithm we also tested higher thresholds up to 25% although we would not expect to apply thresholds of this level in real use.

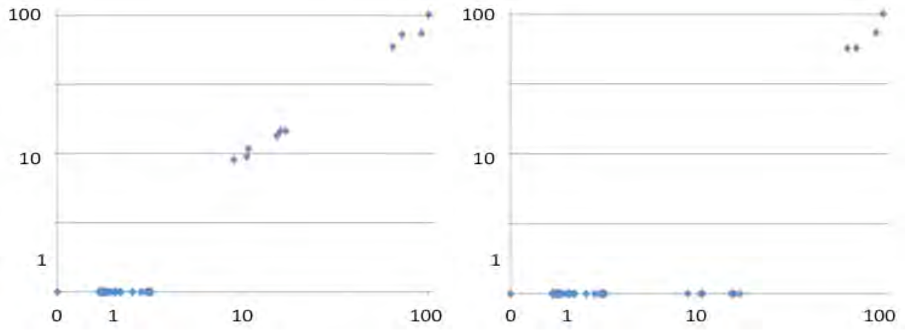
Each plot in Figures 10 and 11 is a log-log plot where each point represents one or more entities (there is heavy over-plotting as many entities have similar activation). The horizontal x-axis is the log of the activation with no threshold and the vertical y-axis is the activation when the threshold is applied (strictly log(activation+1), to allow for zero activation). The latter will never be greater than the activation with no threshold, so the line points will always lie on or below the 45 degree diagonal (the line y=x).

If there were no impact of the threshold at all this would be a perfect diagonal line, but we would expect that as the threshold is introduced some entities would drop in their level of activation. We are particularly interested in the entities with higher activation, if any of these dropped to become low or so that the overall order of activation changed, this would be a potential source of inaccuracy.



(i) threshold = 1

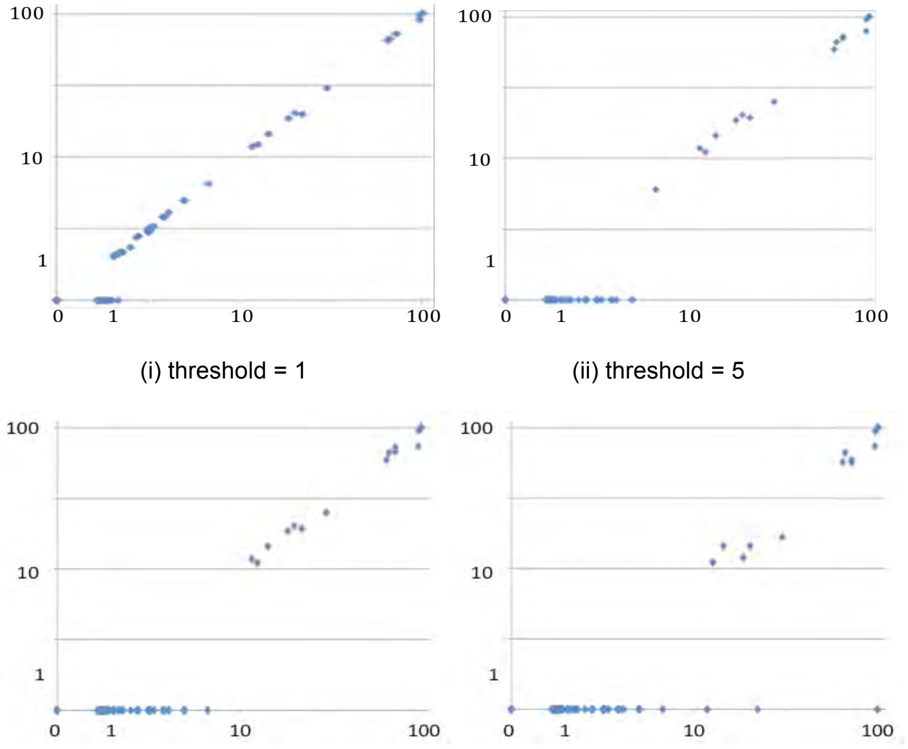
(ii) threshold = 5



(iii) threshold = 10

(iv) threshold = 25

Figure 10. Log-log plots, single activated node, thresholds at 1, 5, 10, 25



(i) threshold = 1

(ii) threshold = 5

(iii) threshold = 10

(iv) threshold = 25

Figure 11. Log-log plots, two activated nodes, thresholds at 1, 5, 10, 25

The results with both one and two initially activated nodes are similar. In both cases the smaller thresholds of 1 and 5 show that those nodes near or below the threshold have their activation cut to zero, whereas those above the threshold are hardly affected. The exact value of thresholds will be a trade-off between obtaining a sufficiently rich context and efficiency; but these seem fairly typical of the values one might choose.

In addition higher values of 10 and 25 have also been plotted (Figure 10 and 11 (iii) and (iv)). The latter is far larger than one would use as it is well above the 'knee' in the typical activation patterns in Figures 8 and 9 and is towards the centre of the output range of the sigmoid (0-100). However, the picture presented by these graphs is still remarkable stable at a threshold of 10 with relatively little impact on the more highly activated entities. It is only at the extreme 25 threshold that we start to see major effects with several of the highly activated entities 'crashing' down to near zero activation (Figure 11.iv). On closer examination it turns out that the affected entities are actually those representing classes (Place, Country, Person). In our implementation of spreading activation, 'instance_of' relations are treated uniformly with other relations. This is largely to allow and email that mentions a class term such as 'cat' to have an impact, so that 'Tom' elsewhere in the email may be more likely to be interpreted as the cartoon animal than the name of a friend. Almost as an accident these classes then tend to get activated due to multiple instances being activated and are more sensitive to the threshold than 'normal' entities..

5.3. *Scaling up to a large-scale data set*

In order to validate the SA algorithm on larger data sets we used the programmes and music data made available by the BBC as part of its Backstage initiative [68]. In particular we used the SPARQL endpoint, which contains approximately 20 million triples [69] organized according to the BBC's programmes and music ontologies [70,71] and is hosted on a Talis platform store [72]. These triples refer to both external resources and also approximately 435,500 internally minted URI entities (372,000 related to programmes, including episodes, series and brands; and 63.500 entities related to music, including solo artists, groups, albums and reviews).

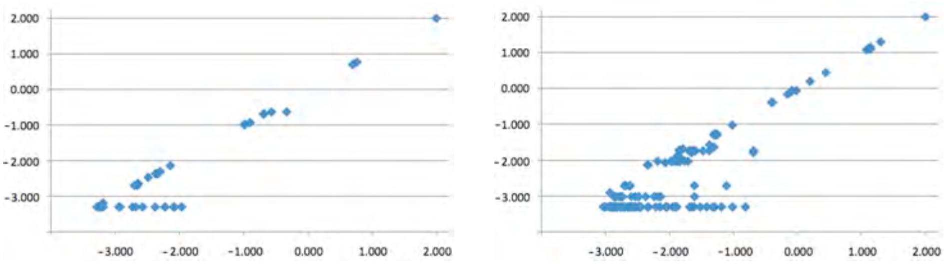
The ontologies used in the Backstage RDF attempt to follow best practice in linked data, for example, adopting standard ontologies such as FOAF and Dublin Core where possible and including 'owl:sameAs' links to dbpedia. It is thus both large enough to act as a realistic test case (orders of magnitude larger than can be used in memory or even traversed in stages) and also is a paradigm of the expected form of future web data.

Although our aim is to integrate the personal ontology with web resources, for the purposes of these scaling experiments we created a dedicated test engine in PHP using the Moriaty library [73] to access the Tails store and parts of the ARC2 library [74] for in-memory graphs. Otherwise the algorithms from section 3 were duplicated as closely as possible, with the exception of the new features needed for dealing with remote cached data.

With the personal ontology we could run algorithms with no thresholds to create a base case, effectively spreading over the entire ontology. For the BBC data set, just like the whole web, we cannot traverse the entire data set. Instead we used very low thresholds (0.01%) as a baseline and then a semi-logarithmic range of thresholds above (0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5). We applied both plain thresholding and also threshold with lookahead as described in section 4.4. We also used a variety of spreading factors, used as the value for parameters A and B from section 3.3.1 and relation weights inversely proportional to fan-out.

Figures 12, 13 and 14 all show log-log plots similar to those produced for the personal ontology data. All of them compare a threshold of 1.0 used as a typical value that might be used in practice with the baseline threshold of 0.01. (Note that like the personal ontology data the log was offset to deal with zero activation, in this case $\log_{10}(x+0.001)$.)

Starting with Figure 12, this shows two spread factors, one more aggressive (0.3) and the other a little less so (0.2). The impact of the spread factor is non-linear hence this is quite a large difference. In both cases it is evident that the higher activation entities (those with activation higher than $\log_{10}(1)=0$) are not affected by the application of the threshold. The data here was seeded from a television episode.

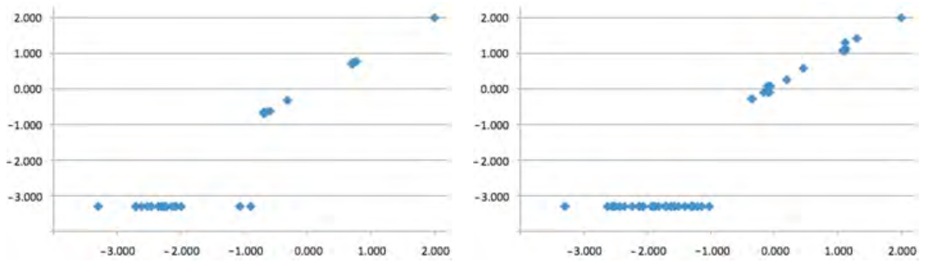


(i) spread factor 0.2

(ii) spread factor 0.3

Figure 12. Using plain threshold BBC programme data

Figure 13 shows the same seed and scale factors, but where the use of lookahead has also been enabled. The lookahead threshold was set at 20% of the entity loading threshold, effectively meaning that triples where the predicate's fan-out was greater than 5 were not included. Like the results for plain thresholding, there is no appreciable alteration in activation levels for those above the threshold.



(i) spread factor 0.2

(ii) spread factor 0.3

Figure 13. Using threshold plus lookahead BBC programme data

Table 1 compares the numbers of triples loaded by plain thresholding and the use with lookahead. As is evident the effect is dramatic reducing the numbers of triples loaded by a factor of around 30:1 for a typical threshold value of 1.0. Although we did not try to optimize the algorithm, the actual spreading activation is fast even for large cached graphs, and the speed are dominated by remote access times. Controlling the number of triples loaded is essential to making this a practical approach. In fact, at the threshold of 1.0, the algorithm including download times was of the order of a second including logging outputs, so fast enough for embedding as context inference for interactive systems.

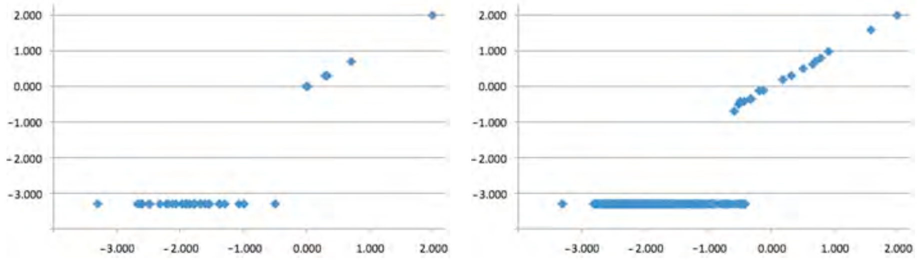
		threshold value								
lkhd	***sf***	*0.01*	*0.02*	*0.05*	*0.1*	*0.2*	*0.5*	*1*	*2*	*5*
no	*0.2*	*1873*	*1869*	*1869*	*1869*	*1655*	*1127*	*1127*	*1127*	*11*
yes	*0.2*	*503*	*93*	*44*	*39*	*39*	*36*	*34*	*11*	*11*
no	*0.3*	*7653*	*2936*	*1873*	*1869*	*1869*	*1853*	*1349*	*1127*	*1127*
yes	*0.3*	*654*	*548*	*310*	*151*	*60*	*46*	*42*	*39*	*11*

Table 1. Number of triples loaded with and without lookahead
Key: lkhd=uses lookahead, sf=scale factor

At an implementation level, we had to simulate the lookahead partially, because SPARQL does not provide a 'COUNT' function like SQL. The code effectively loaded the full data for target entities and then discarded triples with predicates with high fan-out before applying the spreading activation. Even this made a substantial difference in performance as the discarded triples did not contribute to further entity activation, but was still downloading many unnecessary triples. However, SPARQL 2.0 is planned to have counts so it will be possible in future to garner sufficient meta-information to prevent these unnecessary downloads.

Finally Figure 14 shows similar results to those in Figure 13, that is with thresholds and lookahead, but this time applied to seeds connected with music (in fact band members of 'Queen'). The music parts of the BBC Backstage data are more richly interconnected than the programmes data, so this gave us a test of the stability of the algorithm on data with different topological characteristics (even if they are in the same larger data set).

Again the accuracy is good with little change in the high-activation entities. However, do note that we have chosen different spread factors. This is because the more aggressive spread factors (0.4 and 0.3) were fetching very large numbers of triples with the low base line threshold (over 25000 entities were being loaded at threshold of 0.001). While the numbers were manageable at typical thresholds, this does suggest that some form of cap-based self-adjusting thresholds may be appropriate to allow the algorithm to self adjust to different kinds of data.



(i) spread factor 0.1

(ii) spread factor 0.2

Figure 14. Using threshold plus lookahead BBC music data

5.4. *Summary*

In summary, we have seen that the web of data has substantial locality and our reliance on this is a common assumption with other forms of web-scale reasoning. Looking specifically at spreading activation, test data from a populated personal ontology suggests that the working set of highly activated entities is quite small and furthermore that by introducing thresholds at reasonable values, we do not substantively affect the activation profile behaviour of this working set. This has been borne out in larger experiments on a substantial data set representative of best practice in linked data.

6. Discussion

6.1. *Symbolic reasoning over the web – the warm world assumption*

When performing (principally symbolic) reasoning over knowledge bases there is a traditional distinction between the *closed world* assumption (CWA) that acts as if the knowledge available is complete and *open world* assumption (OWA) that assumes the knowledge is partial. For CWA we can assume that if a fact is not in the knowledge base it is not true, whereas for OWA we simply assume it is unknown unless there is an explicit negative fact. Because of this CWA can typically include negation (in the sense of ‘not found’) in reasoning steps, but OWA cannot and universal quantification or instance enumeration is meaningless for OWA.

In the Semantic Web / RDF world, it is typically explicitly stated that the knowledge base is assumed to be partial – that is OWA is assumed to hold, although Kalfoglou et al. [75] suggest that earlier AI planner techniques [76] can be used to establish a *local world assumption* (also known as *local closed world* or LCW), where subsets of the data are known to be complete.

If we have used spreading activation (or indeed any algorithm that brings in data from external sources), then we can perform CWA over the resultant cached ontology *as if* it were the complete model of the world. This is reasoning over the currently relevant or salient knowledge. We call this the *warm world assumption,* WWA.

Warm world reasoning is defeasible as new knowledge may later be brought in that changes previous inferences. However, this is similar to human reasoning about the world, which is based on what we know at any moment in time.

In some ways WWA is similar to the LarKC sampling approach shown in Figure 2 [3,39]. In both cases reasoning is applied to a sample of the full data. However, whilst LarKC assumes a random sample, we are effectively having something closer to a snowball sample. The random sample makes it easier to derive probabilistic or asymptotic properties of the reasoning, but WWA is more likely to be operating on relevant or popular parts of the overall data, and also to have interesting inter-entity relations.

The quality of reasoning from WWA or any sample-based technique, will depend on the kinds of reasoning rules. We do not have sufficient experience to give strong advice on this, but there are some obvious cases where we expect particularly good or poor.

Some rules have good locality, for example:

```
Person(X): lives_in(X,P1) AND part_of(P1,P2) => lives_in(X,P2)
```

Here, if a person "Alan" is in the working set for WWA and is sufficiently active, then connected entities such as the place Alan lives ("Tiree") and places it is within ("Scotland"), will also be in the working set. So, the above rule is likely to work if any of the entities involved are highly active.

On the other had, where there is a quantifier existential with a single or small number of satisfying instances, deductions regarding these are unlikely to be successful. For example,

```
EXISTS( Person(X), Country(C), is_king_of(X,C) AND number_of_wives(X,"6") )
```

Here, we would not expect to find "Henry VIII" when the activation is related to a semantic web paper. However, if the activated topic is related to monarchs of England, then the inference is likely to be successful. So with WWA, quite reasonably, we expect poor performance for rules that are not connected with the topic under scrutiny.

In between the two are cases where the inference would establish links between the entities in the working set and some other island of data; for example, links based on shared property values:

```
Student(X),Citizen(Y): email_of(X,E) AND email_of(Y,E) => same_as(X,Y)
```

These are important as they are clearly relevant if we have either a student or citizen entity, but if the students and citizens sit in otherwise isolated data islands, we risk missing the linkage. In this cases it would be good for the symbolic reasoning to influence the sub-symbolic processing.

6.2. *Using reasoning to influence caching and/or activation*

This reasoning process could be carried out independent of the activation process that brings in data, but could also influence the process of choosing what to fetch.

First, we may directly fetch data that is sufficiently instantiated. For example, suppose we have a reasoning rule:

```
IF < City c, capital_of, Country X > AND X,population >1 million THEN …
```

If we apply this to a known city 'Athens', but do not have the fact <'Athens', *capital_of*, 'Greece'> in our local ontology, then by pure CWA this would evaluate to false. However, the condition triple is partially instantiated <'Athens', *capital_of*, Country X>, so we could access a relevant source (e.g. Geonames) at this point in the reasoning. However, we would not do so if we only had one element of the triple instantiated, or of the fan-out of the relation was very large. This kind of expansion of the reasoning to include external knowledge sources is similar to those proposed for pure symbolic web-scale reasoning approaches such as [48]. However, based on the warm world assumption, we could also accept queries such as:

```
EXISTS( Person(X), lives_in(X,"Tiree") AND works_with(X,"Costas") )
```

Answering such a query would not involve finding all Person entities on the web (perhaps possible using a semantic web search engine such as Swoogle, but too large to download). Instead it is taken to mean "all entities of type Person who are known about in the personal ontology or have sufficiently high activation to be cached locally".

Alternatively, or in addition, if the process of following symbolic reasoning repeatedly encounters certain entities, then this could add activation to those entities. This would mean that if an entity is critical in multiple parts of reasoning, it may be fetched due to one of the activation-based fetch rules.

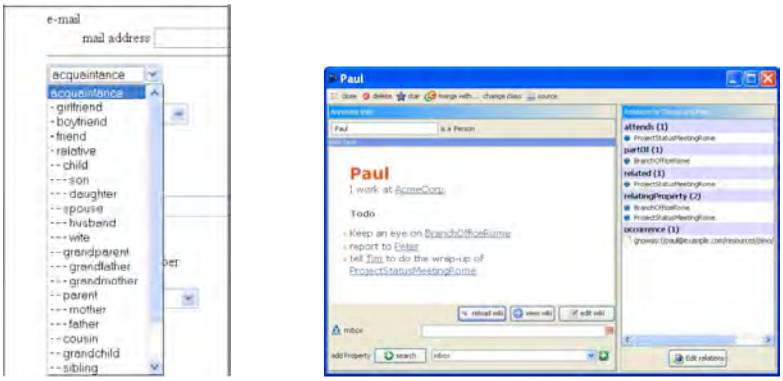
Of course, when new entities and triples are fetched, this may affect previous inferences. We may then want to use forward incremental reasoning to modify past inferences based on the new data using mechanisms similar to those proposed for Active Triple Spaces [77].

6.3. *Provenance and the user*

When entities and relations are fetched from external sources or inferred based on reasoning rules, some form of provenance needs to be maintained. This is important algorithmically in order to periodically refresh dynamic data, but also so that we can effectively present the ontology to the user. If the user is browsing their ontology there is a danger that their carefully entered data will get swamped in imported data. No matter how relevant this will be at best disorienting and at worst frustrating, users need a *deterministic ground* [78], parts of the interface they can trust *not* to change even though other parts adapt. This doesn't mean automatically fetched data or even inferred data cannot be presented to the user, but it needs care.

For form-fill suggestions such as city names, this is less of a problem as the set of candidates is likely to be small, although even there we may wish to present personal ontology items differently, maybe a different font style. However, in PIM system we may wish the user to be able interact with their personal ontology; for example this is done in our own ontology Profiler [13] and the Gnowsis ThingEditor [79] (see Figure 15). In these we do not want the user suddenly exposed to vast numbers of classes and instances that have been brought in from the web. Indeed, in long-term studies of Gnowsis, users seemed to prefer well-trodden paths through their personal ontology, not searches or alternatives found by the system [79]. However, whilst we do not want automatically inferred or fetched data to take over the user interface, the user might reasonably expect to be able to browse such data, especially if it is deemed especially relevant due to high activation and becoming part of interaction. Moreover the users may want to 'claim' such data as their own, still externally sourced, but very much part of their personal information.

We will not attempt to solve these user presentation and interaction issues here, but note that core to being able to present data appropriately is that provenance be recorded.



Paul is a Person

Paul

I work at AcmeCorp.

Todo

- Keep an eye on BranchOfficeRome
- report to Peter
- tell Tim to do the wrap-up of ProjectStatusMeetingRome

(i) Ontology Profiler [13]

(ii) Gnowsis Thing Editor [79]

Figure 15. User interfaces for browsing person ontologies

6.4. *Collaboration and sharing*

Our main focus in this paper has been personal ontologies and a single user's interactions. However, one of the great success of the web over recent years has been the social web: explicit social networking in sites such as Facebook; implicit recommender systems [80] such as used by Amazon; and folksonomies based on explicit individual tagging, but leading to emergent shared vocabularies, which O'Reilly identified as one of the key features of Web2.0 [81].

Spreading activation is effectively about allowing an automated assistant to share some level of context with the user. Some years ago, before the advent of Web2.0, one of the authors wrote about the idea of the 'web sharer' [82], a vision of the Internet as place of sharing not just publishing, which has now become reality. In comparing the Internet to the physical world it said, "Physical human interaction is about sharing words and things within a shared context." The 'shared context' here referred to shared human–human context, creating places for sharing, as found in sites such as Facebook. However, having an automatically captured context raises the question as to whether this can be used to enable some form of emergent, shared computational context (see figure 16).

If context inference were restricted to personal ontologies, then the scope for this is limited as many of the entities will only be of interest to individuals. However, once we include shared resources such as corporate or workgroup data stores, or the web itself, then patterns of activation can be used as indications of emerging interests across groups of friends, work colleagues or entire communities.

At a simple level it would be possible to use the average LTA of a work group such as a project team, to feed into an individual's spreading activation alongside their own LTA and MTA. This would mean, for example, that a new employee would find resources relevant to their organization more highly weighted than less relevant ones. Similarly when receiving an email, it would be possible to capture some of the sender's MTA as this relates to the recent activity of the sender when writing the email; this would mean that shared entities (e.g. films or music) that were active for the sender would be more likely to be suggested to the recipient when acting in response to the email.

There are potential implications for privacy and security; even if this is only applied to shared entities, then the activations created may still carry unexpected information. For example, a group of bikers might notice that "Bridget Jones" was highly activated alongside "Harley Davidson" ... eliciting a search for the closet 'chic flick' addict.

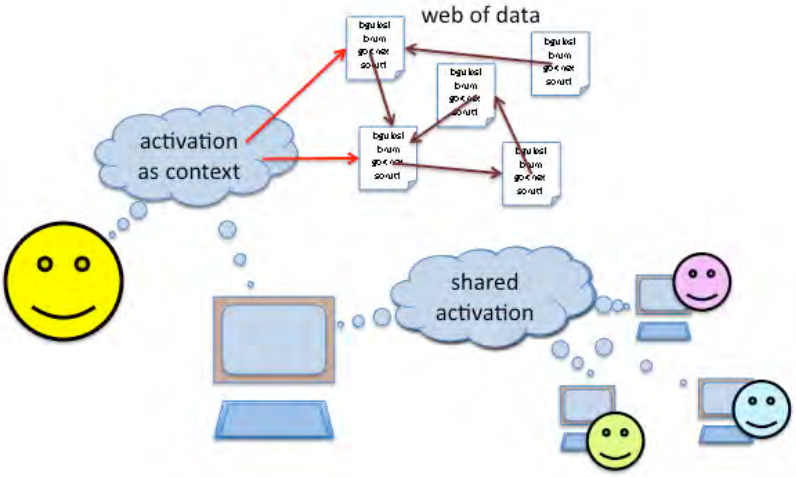


Figure 16. Sharing activation to give shared context

7. Conclusions and Further Work

In this paper we have described how our existing use of spreading activation to model context in personal ontologies can be extended to allow the inclusion of larger remote information repositories, including the entire web of data. The basic approach is to draw in information from remote sources based on the activation of entities already held in memory. The mechanisms depend on the assumption that the working set of highly activated entities will be small and that excluding entities with sufficiently low activation does not adversely affect the effectiveness of the resulting pattern of activation. Data drawn from published sources and our own experiments on a populated personal ontology are encouraging and suggest that these assumptions are likely to be valid. Furthermore tests of the algorithms on a single large linked-data dataset give good evidence of web-size scaling as the BBC Backstage dataset is effectively unbounded compared to the in-memory cache.

The nascent state of existing linked data sets means we cannot fully predict the properties of the long-term web of data from those currently available. However, we do wish to link to this existing linked data both to validate our approach further and also to begin to make practical use of this additional data during user interaction.

We have discussed several potential issues arising from this work and also directions in which it can be developed, both in terms of its impact on users and its internal algorithms. In particular, we have outlined the warm world assumption: treating activated entities as the universe during reasoning.

Various authors including ourselves have proposed that the human brain may be used as a metaphor or an inspiration for developments on the web [1,53,83,84,85]. While our work has very practical roots, we do look repeatedly at features of human intelligence to suggest appropriate automated methods. Even the proposals for warm world reasoning are similar to the way humans blend associative and deductive reasoning. Our own emphasis is on assisting humans in day-to-day activities; both in this and in other application domains, the deliberately human-like traits of our approach offer the prospect of truly web scale reasoning.

Acknowledgments

The DELOS EU Network of Excellence supported early parts of this work. We thank particularly colleagues working on the TIM project within DELOS where the concepts elaborated in the paper were first developed.

References

[1] A. Katifori, C. Vassilakis and A. Dix, Ontologies and the Brain: Using Spreading Activation through Ontologies to Support Personal Interaction. Cognitive Systems Research (in press) (2009).

[2] Linked Data - Connect Distributed Data across the Web, linkeddata.org administered by Tom Heath on behalf of the Linked Data community, http://linkeddata.org/

[3] C. Bizer, T. Heath, T. Berners-Lee, Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems (in press) (2009).

[4] D. Fensel, F. van Harmelen (2007). Unifying Reasoning and Search to Web Scale, *IEEE Internet Computing* 11(2) (2007) 94–96.

[5] G. Anadiotis, S. Kotoulas, and R. Siebes, An architecture for peer-to-peer reasoning, *in Proc. of the First Int. Workshop "New forms of reasoning for the Semantic Web: scalable, tolerant and dynamic", co-located with ISWC 2007 and ASWC 2007*, Vol-291, Busan, Korea, 2007, http://CEUR-WS.org/Vol-291/

[6] A. Qasem, D. A. Dimitrov, and J. Heflin. ISENS: A Multi-ontology Query System for the Semantic Deep Web, *in Proc. of Workshop on The Semantic Web meets the Deep Web*, IEEE CEC'08 and EEE'08, Washington DC, 2008.

[7] R. T. Gruber, A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition, Special issue: Current issues in knowledge modeling* 5(2) (1993) 199-220

[8] J. Trajkova, S. Gauch, Improving Ontology-based User Profiles, *in Proc. of RIAO 2004*, University of Avignon (Vaucluse), France, 2004, 380-389

[9] S. Gauch, J. Chaffee, and A. Pretschner, Ontology-Based User Profiles for Search and Browsing, *Web Intelligence and Agent Systems*, Vol. 1, No. 3-4. (2003), pp. 219-234.

[10] V. Katifori, A. Poggi,. M. Scannapieco, T. Catarci, Y. Ioannidis, Y. OntoPIM: how to rely on a personal ontology for Personal Information Management, *in Proc. of the 1st Workshop on The Semantic Desktop*, Galway, Ireland, 2005.

[11] L. Sauermann, The Gnowsis Semantic Desktop for Information Integration, *in Proc. of the 3rd Conference Professional Knowledge Management*, Kaiserslautern, Germany, 2005.

[12] P.-A. Chirita, R. Gavriloai, S. Ghita, W. Nejdl, R. Paiu, Activity Based Metadata for Semantic Desktop Search, *in Proc. of the 2nd European Semantic Web Conference*, Heraklion, Greece, 2005.

[13] M. Golemati, A. Katifori, C. Vassilakis, G. Lepouras, C. Halatsis, Creating an Ontology for the User Profile: Method and Applications, *in Proc. Of the First RCIS Conference*, Ouarzazate, Morocco, 2007.

[14] A. Katifori, C. Vassilakis, I. Daradimos, G. Lepouras, Y. Ioannidis, A. Dix, A. Poggi, T. Catarci, Personal Ontology Creation and Visualization for a Personal Interaction Management System, *in Proc. of PIM Workshop*, CHI 2008, Florence, Italy, 2008.

[15] A. Katifori, C. Vassilakis, A. Dix, I. Daradimos, G. Lepouras, *Spreading activation user profile ontology*, University of Athens, Technical Report, http://oceanis.mm.di.uoa.gr/pened/?category=pub#ontos

[16] A. Dix, Tasks = data + action + context: automated task assistance through data-oriented analysis, *in Proc. of Engineering Interactive Systems 2008: Second Conference on Human-Centered Software Engineering, HCSE 2008 and 7th International Workshop on Task Models and Diagrams*, *TAMODIA 2008*, Pisa, Italy, 2008.

[17] E. Rukzio, A. Schmidt, H. Hussmann, Privacy-enhanced Intelligent Automatic Form Filling for Context-aware Services on Mobile Devices, *in Proc. of workshop on Artificial Intelligence in Mobile Systems 2004 (AIMS 2004), in conjunction with UbiComp 2004*, Nottingham, UK, 2004.

[18] W3C draft, Client Side Automated Form Entry, W3C Working Draft WD-form-filling-960416, http://www.w3.org/TR/WD-form-filling.html

[19] A. Dix, A. Katifori, A. Poggi, T. Catarci, Y. Ioannidis, G. Lepouras and M. Mora, From Information to Interaction: in Pursuit of Task-centred Information Management, i*n Proc. of DELOS Conference 2007*, Pisa, Italy, 2007.

[20] J. R. Anderson, A Spreading Activation Theory of Memory, *Journal of Verbal Learning and Verbal Behaviour*, 22 (1983) 261-295.

[21] F. Crestani, Application of spreading activation techniques in information retrieval, *Artificial Intelligence Review* 11(6) (1997) 453–482.

[22] F. Crestani, Retrieving documents by constrained spreading activation on automatically constructed hypertexts, *in Proc. of EU- FIT 97- Fifth International Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, 1997, pp. 1210-1214.

[23] W. Liu, A. Weichselbraun, A. Scharl, and E. Chang, Semi-Automatic Ontology Extension Using Spreading Activation, *Journal of Universal Knowledge Management*, 0(1) (2005), 50 – 58.

[24] R. G. Xue, J. H. Zeng, Z. Chen, Y. W. Ma, W. Xi, W. Fan and Y. Yu, (2004). Optimizing Web Search Using Web Click-through Data. *in Proc. of ACM Thirteenth Conference on Information and Knowledge Management (CIKM)*, Washington D.C., U.S.A., 2004, pp. 118-126

[25] M. Hasan, A Spreading Activation Framework for Ontology-enhanced Adaptive Information Access within Organisations. *in Proc. of the Spring Symposium on Agent Mediated Knowledge Management (AMKM 2003)*, Stanford University, California, USA, 2003.

[26] J. J. Hopfield, Neural networks and physical systems with emergent collective computational properties, *in Proc. of the National Academy of Sciences of the USA*, 79 (1982), pp. 2554 - 2588.

[27] A. Katifori, C. Vassilakis and A. Dix, Using Spreading Activation through Ontologies to Support Personal Information Management, *in Proc. of Common Sense Knowledge and Goal-Oriented Interfaces, held in conjunction with the 2008 International Conference on Intelligent User Interfaces (IUI 2008)*, Canary Islands, Spain, 2008.

[28] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American*, May 2001.

[29] About Freebase, MetaWeb Technologies, http://www.metaweb.com/about/

[30] Google Spreadsheets APIs and Tools, http://code.google.com/apis/spreadsheets/overview.html

[31] G. Price, C. Sherman, The Invisible Web: Uncovering Information Sources Search Engines Can't See, (CyberAge Books, 2001).

[32] B. He, M. Patel, Z. Zhang, and C. K. Chang, Accessing the Deep Web: A Survey, *Communications of the ACM (CACM)* 50(2) (2007) 94–101.

[33] G. P. Ipeirotis, L. Gravano and M. Sahami, Probe, count, and classify: categorizing hidden web databases, *SIGMOD Rec*. 30(2) (2001) 67-78.

[34] J. P. Bigham, A. C. Cavender, R. S. Kaminsky, C. M. Prince and T. S. Robison, Transcendence: Enabling a Personal View of the Deep Web, *in Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI 2008)*, Canary Islands, Spain, 2008.

[35] D. Gibson, K. Punera, and A. Tomkins, 2005. The volume and evolution of web page templates, *in Special interest Tracks and Posters of the 14th ACM Int. Conference on World Wide Web (WWW '05)*, Chiba, Japan, 2005.

[36] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, A. J. Tomlin, and Y. J. Zien, SemTag and seeker: bootstrapping the semantic web via automated semantic annotation, *in Proc. of the 12th ACM Int. Conference on World Wide Web (WWW '03)*, Budapest, Hungary, 2003.

[37] Google Sets, http://labs.google.com/sets

[38] LarKC: The Large Knowledge Collider, http://www.larkc.eu/

[39] D. Fensel, F. van Harmelen, B. Andersson, P. Brennan, H. Cunningham, E. Della Valle, F. Fischer, Z. Huang, A. Kiryakov, T. Kyung-il Lee, L. Schooler, V. Tresp, S. Wesner, M. Witbrock and N. Zhong (2008). Towards LarKC: A Platform for Web-Scale Reasoning, *in Proc. of IEEE second Int. Conference on Semantic Computing (ICSC 2008),* Santa Clara, CA, USA, 2008, pp.524–529

[40] H. Simon, *Models of Man*, (Wiley, 1957).

[41] R. Motwani, and P. Raghavan, 1996. Randomized algorithms, *ACM Comput. Surv.* 28(1) (1996), 33-37.

[42] C. Gomes, J. Hoffmann, A. Sabharwal, and B. Selman. From sampling to model counting. In Proc. of International Joint Conferences on Artificial Intelligence (IJCAI '07), Hyderabad, India, 2007, p.p. 2293–2299.

[43] C. Morbidoni , A. Polleres , and G. Tummarello, Who the FOAF knows Alice? A needed step towards Semantic Web Pipes, *in Proc. of the First Int. Workshop "New forms of reasoning for the Semantic Web: scalable, tolerant and dynamic", co-located with ISWC 2007 and ASWC 2007*, Vol-291, Busan, Korea, 2007, http://CEUR-WS.org/Vol-291/

[44] S. Ceri, E. Della Valle, D. Fensel, F. van Harmelen. R. Studer, 1*st Int. Workshop on Stream Reasoning*, Heraklion, Greece, 2009, http://streamreasoning.org/

[45] R. Wallis, Bigfoot - An initial tour, Talis Platform User Guide, 2007 http://www.talis.com/tdn/platform/user/bigfoot/tour

[46] J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, *in Proc. of Sixth Symposium on Operating System Design and Implementation (OSDI '04)*, San Francisco, CA, U.S.A., 2004, http://labs.google.com/papers/mapreduce.html

[47] A. Hogan, A. Harth, and A. Polleres. Scalable authoritative OWL reasoning for the web. *International Journal on Semantic Web and Information Systems*, 5(2), April-June 2009.

[48] A. Qasem , D. Dimitrov, and J. Heflin, Efficient Selection and Integration of Data Sources for Answering Semantic Web Queries, *in Proc. of the First Int. Workshop "New forms of reasoning for the Semantic Web: scalable, tolerant and dynamic", co-located with ISWC 2007 and ASWC 2007*, Vol-291, Busan, Korea, 2007, http://CEUR-WS.org/Vol-291/

[49] S. Brin, and L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, *in Proc. of Seventh ACM International World-Wide Web Conference (WWW '98)*, Brisbane, Australia, 1998.

[50] C. R. Atkinson, M. R. Shiffrin, Human memory: A proposed system and its control processes. in The psychology of learning and motivation, ed. K.W. Spence and J.T. Spence, vol. 8. (Academic Press, 1968).

[51] G. A. Miller, The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, *The Psychological Review*, 63(2) (1956) 81-97.

[52] A. K. Ericsson and W. Kintsch, Long-term working memory, *The Psychological Review*, 102(2) 1995 211-245.

[53] A. Dix, The brain and the web: intelligent interactions from the desktop to the world. In *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems, IHC'06*, vol. 323. (2006) 142. DOI=10.1145/1298023.1298080

[54] The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/

[55] M. T. Mitchell, Machine Learning, (McGraw-Hill, 1997).

[56] T. Lømo, The discovery of long-term potentiation. *Philosophical Transactions: Biological Sciences*. 358(1432) (2003) 617-620.

[57] A. Katifori. *Preliminary Evaluation Example of the Spreading Activation Algorithm*. University of Athens, Technical Report, http://oceanis.mm.di.uoa.gr/pened/?category=pub#ontos

[58] G. Salton and M.J. Mc Gill, *Introduction to modern information retrieval*. Mc-Graw-Hill, Singapore (1987).

[59] A. Cheyer, Ja. Park, R. Giuli, IRIS: Integrate. Relate. Infer. Share, *in Proc. Workshop on the Semantic Desktop: Next Generation Personal Information Management and Collaboration Infrastructure (ICSC 2005*), Galway, Ireland, 2005.

[60] A Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J. Wiener, Graph structure in the Web, Computer Networks 33(1-6) (2000) 309-320.

[61] DESCRIBE Query, SPARQL Query Language for RDF. W3C Recommendation (2008) http://www.w3.org/TR/rdf-sparql-query/#describe

[62] A. S. Tanenbaum, *Modern Operating Systems* (3rd edition). (Prentice Hall, 2007)

[63] A. Harth, A. Hogan, J. Umbrich, S. Decker, Building a Semantic Web Search Engine: Challenges and Solutions, *in Proc. of XTech 2008: "The Web on the Move"*, Dublin, Ireland, 2008, http://2008.xtech.org/public/schedule/detail/477

[64] Swoogle Statistics. No longer available, but as cited in [4]. Originally at http://swoogle.umbc.edu/2005/modules/Swoogle Statistics/images/figure5-2004-09.png

[65] L. Ding, L. Zhou, T. Finin, and A. Joshi, How the Semantic Web is Being Used: An Analysis of FOAF, in Proc. of the 38th International Conference on System Sciences, Hawaii, U.S.A., 2005.

[66] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, January 1998, pp. 668–677.

[67] H. Stuckenschmidt and M. Klein, Structure-Based Partitioning of Large Concept Hierarchies}, International Semantic Web Conference, LNCS 3298, pp. 289-303, http://www.informatik.uni-trier.de/~ley/db/conf/semweb/iswc2004.html#StuckenschmidtK04

[68] BBC Backstage. British Broadcasting Corporation, dated 2004-2005, Accessed 6/7/2009. http://backstage.bbc.co.uk/

[69] L. Dodds, Understanding the Big BBC Graph, dated 11th June 2009. Accessed 10/08/2009. http://blogs.talis.com/n2/archives/569

[70] Y. Raimond, P. Sinclair, N. Humfrey, M. Smeturst, Programmes ontology, British Broadcasting Corporation, dated 17th April 2009. Accessed 6/7/2009. http://purl.org/ontology/po/2009-04-17.shtml

[71] F. Giasson, Y. Raimond, Music Ontology Specification, British Broadcasting Corporation, dated 5th February 2007. Accessed 6/7/2009. http://purl.org/ontology/mo/

[72] Talis Platform. Talis. Accessed 6/7/2009. http://www.talis.com/platform/

[73] I. Davis. Moriarty, n2 Wiki. Iand 09:00, dated 2 October 2007. accessed 6/7/2009. http://n2.talis.com/wiki/Moriarty

[74] B. Nowack, ARC2: Easy RDF and SPARQL for LAMP systems. Semsol. accessed 6/7/2009. http://arc.semsol.org/

[75] Kalfoglou, Y., Alani, H., Schorlemmer, M. and Walton, C. (2004) On the emergent Semantic Web and overlooked issues. In: 3rd International Semantic Web Conference, November 2004, Hiroshima, Japan.

[76] K. Golden, O. Etzioni, and D. Weld. Omnipresence without omnipresence. In Proc. of 12th Nat. Conf. on Artificial Intelligence(AAAI'94), 1994.

[77] V. Tanasescu, Differences + Triple Spaces = Active Triple Spaces, *in Proc. of the First Int. Workshop "New forms of reasoning for the Semantic Web: scalable, tolerant and dynamic", co-located with ISWC 2007 and ASWC 2007*, Vol-291, Busan, Korea, 2007, http://CEUR-WS.org/Vol-291/

[78] A. Dix, J. Finlay and J. Hassell, Environments for cooperating agents: Designing the interface as medium, *in CSCW and Artificial Intelligence*, eds. J. Connolly and E. Edmonds, (Springer Verlag, 1994) pp. 9-26.

[79] L. Sauermann and D. Heim, Evaluating Long-Term Use of the Gnowsis Semantic Desktop for PIM, *in Proc. of the 7th international Conference on the Semantic Web*, Karlsruhe, Germany, 2008, 467-482.

[80] P. Resnick and H. R. Varian (Eds.), Special Issue on Recommender Systems, *Communications of the ACM*, 40(3) (1997) 56–89

[81] T. O'Reilly, What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software, (2005) http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

[82] A. Dix, The web sharer vision. Working Paper. aqtive limited, (1999) http://www.hiraeth.com/alan/ebulletin/websharer/vision-web-sharer.html

[83] G. Mayer-Kress and C. Barczys, The global brain as an emergent structure from the worldwide computing network, *The information society* 11(1) (1995) 1-28.

[84] K. Lerman , The Web-Brain Hypothesis, University of Southern California (1998) http://www.isi.edu/~lerman/etc/brain.html

[85] N. Zhong, Towards Human Level Web Intelligence: A Brain Informatics Perspective, *Pattern Recognition and Machine Intelligence*, LNCS 4815 (2007) p. 311, DOI: 10.1007/978-3-540-77046-6_38