# Context-Aware Retrieval for Ubiquitous Computing Environments

Gareth J. F. Jones[1] and Peter J. Brown[2]

[1] School of Computing, Dublin City University
Dublin 9, Ireland
Gareth.Jones@computing.dcu.ie
[2] Department of Computer Science, University of Exeter
Exeter EX4 4PT, United Kingdom
P.J.Brown@exeter.ac.uk

**Abstract.** Mobile and ubiquitous computing environments provide a challenging and exciting new domain for information retrieval. One of the challenges is to provide relevant and reliable information to users often engaged in other activities or to agents acting on behalf of the user. We believe that identification of relevant information can be achieved by integration of existing methods from information retrieval and context-aware technologies. Making use of this retrieved information may be facilitated by contributions from human-computer interaction studies and agent technology to determine how and when to deliver the information to the user or how best to act on the user's behalf.

## 1 Introduction

A key feature of the growth in computing networks is the accompanying rapid expansion in the availability of online information. This is currently seen most obviously in the World Wide Web (WWW), which gives users the ability to access online information from anywhere in the world very rapidly. However, like networked computing itself, the current availability and utilization of online information is far from realising its full potential. The rapid developments in networked computing have the potential to make availability and exploitation of information a fundamental component of ubiquitous computing environments. Information may appear in many forms: within natural language documents in various media; from databases of facts; or aggregated from low-level sensors. Increased bandwidth and developments of mobile computing mean that it will soon be possible to access online material available from the WWW from almost anywhere on earth whatever media it originates in. Whatever source the information is contained within, wherever it is to be delivered, a fundamental issue is how to identify information relevant to an individual user, and, we argue, for ubiquitous computing how best to make use of *this* information for *this* user in their current *context*.

This paper explores issues in context-aware retrieval for mobile and ubiquitous computing. Our analysis sets out the modes of information delivery in networked model computing and the factors affecting the users of these devices and their interaction with the information and the device delivering it. We also describe our proposed methods to improve retrieval effectiveness in this environment in terms of retrieval accuracy and user satisfaction. The primary focus of our current work is the integration of technologies from information retrieval and context-awareness to create systems for reliable and efficient delivery of information in networked context-aware environments.

At present the standard view of accessing documents from the WWW is to download them from their server onto a desktop computer. This picture is currently developing to incorporate download to networked PDAs and mobile phones. Emerging display technologies such as advanced headup displays embedded in eye glasses enable information to be displayed to a user in an augmented view of the world. In order to locate documents of interest users frequently make use of search engines such as *Google*. However, current search engines take no account of the individual user and their personal interests or their physical context. The development of personal networked mobile computing devices and environmental sensors means that personal and context information is potentially available for the retrieval process. We refer to this extension of established information retrieval (IR) as *context-aware retrieval (CAR)* [2] [13]. The objective of incorporating contextual information into the retrieval process is to attempt to deliver information most relevant to the user within their current context. As such we can see the retrieval process as embedded in a context-aware environment, and if it can be made sufficiently "intelligent", as pervasive to the user's world experience. Our interest is mainly in the incorporation of physical context data into the retrieval process. This process may involve personalization of the retrieval process in combination with context-awareness, but this need not be the case.

Retrieval to traditional visible computing devices is only one possibility: in a ubiquitous computing environment the information might be delivered to an *agent* [16] acting on behalf of the user or even on behalf of the institution that owns the environment. The ideal agent could (a) perform actions automatically if there is no need to consult the user; (b) summarise or coalesce documents before presentation; (c) decide when to deliver the retrieved information to the user, e.g. if it is marked as highly relevant, it should be delivered immediately, interrupting whatever they are currently doing; and (d) learn from users how its performance can be improved. All of this puts added demands on a retrieval engine delivering information.

The introduction of information delivery not directly controlled by the user introduces the ideas of *proactive* information retrieval, where a device may automatically initiate a request to a search engine, or may trigger information when the user enters a certain context.

Another important use of context for retrieval within ubiquitous computing is to determine the manner and timing of any information passed to the user. Sensors connected to the user and their environment can enable the user's current activities to be determined[1], thus allowing the retrieval devices to assess whether it is appropriate or safe to disturb the user at a given time. A further limitation of course is that, since the information is based on the user's context, it should be delivered in a timely fashion, since the context may change — it will often be useless to deliver information about a situation the user has just left.

Successful and effective CAR for ubiquitous computing environments may potentially incorporate work from a number of other established and emerging fields, including: human-computer interaction, wearables, agents and wireless networks. This paper introduces the relevant technologies from these areas, explores their integration, and outlines our current research prototype for the investigation of context-aware retrieval. The first three-quarters of the paper represents an ideas paper, building on existing work. However at Section 7 the paper changes gear, and the focus is largely on our own work in building a CAR system, and developing ideas of context-aware caching; a reader with knowledge of the field can read these later Sections independently of the rest, though it is best to read Section 4.1 too.

In detail the paper is organised as follows: Section 2 explores context and its application in context aware retrieval; Section 3 outlines the pertinence of established techniques from information retrieval and information filtering; Section 4 analyses the nature of change in context, and, in Section 4.1, introduces a piece of our relevant work; Section 5 outlines relevant methods for personalisation; Section 6 looks at the use of agents in ubiquitous retrieval application; Section 7, which starts the change of gear, describes some of our practical work that evaluates the effectiveness of these techniques, and Section 8 looks at ubiquitous and timely availability.

## 2 Context and Context-Aware Retrieval

A user's context consists of the their present state, their previous states (history) and their predicted future states (taken from extrapolating past states and/or from future events captured in a diary[9]); this can be enhanced by the contexts of other, similar or related, humans and other objects, or even by the context of information itself. We are primarily interested in physical elements of the user's context, although for information delivery these cannot be separated from the user's personal or cognitive context in a simple way, and we do not seek to make a sharp distinction between these in this discussion.

Mobile applications are the prime field for CAR. This is for three reasons. Firstly information is now being made available in situations it was not available in before. Secondly a mobile user is often in an unfamiliar environment and needs information about that environment. Thirdly, following on from the second point, this is an especially favourable case to use context to help select the information that is needed. Obviously in mobile applications location is a key part of the context. We believe, however, that retrieval is much more effective if the context

[1] This is of course difficult and possible granularity may vary from determining exactly what the user is doing to merely assessing that they are busy or unoccupied.

is richer than just location, and includes fields such as temperature, objects nearby, user's current interests (and even emotional state), etc. A context used to aid retrieval can also usefully include fields that may be considered as aspects of the user model.

CAR is part of the infrastructure needed by a range of applications that detect and exploit context. Such applications are currently in their infancy, particularly if we only consider those that are products rather than research prototypes. This paper attempts to look beyond the needs of current applications, and identify the properties of CAR needed to support the potential applications of the future. We concentrate on issues of retrieval: we assume the existence of a communications infrastructure and of sensors where we need them; we also assume there is an acceptable policy for the privacy of personal information.

Context can also be associated with each of the documents that are candidates for retrieval. Thus a document may have contextual fields representing an associated location or a temperature: an outdoor cafe, for example, is only suitable at certain temperatures. Sometimes these contextual fields are part of the explicit mark-up of a document, and sometimes they need to be derived from, for example, the textual content of the document. A central task of CAR is to match the context of the user with that of each available document.

Documents may also be associated with contextual matter of a different nature to the user's context. An example would be a contextual field that measured the authority of a document [7], in terms of the status of its author, the number of citations or links to it, its revision history, etc. This extra information, though not directly involved in the matching process, can still be used to improve the quality of the material delivered to the user; again we discuss these issues later.

### 2.1 Retrieval Paradigms

CAR is related to the well-established fields of information retrieval (IR) and information filtering (IF). IR and the related technology of IF are concerned with the finding of information, often in the form of text documents, which are in some sense *about* a topic that a user is interested in. Both are concerned with satisfying the user's underlying information need. The user typically expresses their information need as some form of search *request* (sometimes referred to as a *query* or as a *profile*, see later), which is then matched against the available documents. Information is conventionally retrieved from a *collection* of discrete documents. Each document may be sub-divided into *fields*. These fields may be textual, such as title, author, keywords, and the full text of the paper. Alternatively they may be of other data types which are part of the document or accompanying metadata, e.g. numbers, locations, dates, images. The retrieval task is to deliver the documents that best match the current query; each retrieved document may be accompanied with a score that gives a weighting of how well it matches. We distinguish two CAR paradigms as follows:

- *interactive*: A user initiated request is combined with the current context to derive a retrieval query, which is then applied to the document collection in the standard manner used in IR.

– *proactive*: Some or all of the documents in the document collection contain a triggering condition, and when this matches the user's current context the document is supplied to the user. This matching process may include textual fields in both the document and user search profile. This has parallels with IF; the triggering condition has the role of the profile, and the current context acts as the current document; when the current context changes, a new current document is derived and a new retrieval takes place. One difference with IF is that the triggering conditions (profiles) are specified by the provider of the document, not by the user, and apply to all users.

A crucial property of many context fields is that they are *continuous*: as the user's context changes new information may need to be retrieved. Such continuous applications normally require fast retrieval, so that the user has the illusion that new information arrives immediately there is any change in their context. This is absolutely different from the 'one-off' nature of traditional information retrieval requests, and presents many research challenges.

### 2.2 Retrieval Environment

Two key properties of retrieval are *precision* and *recall*. Precision is measured as the proportion of documents retrieved which are relevant, whereas recall is the proportion of the available relevant documents that have been retrieved. With context-aware applications the user is often mobile and frequently involved in other tasks. When a retrieved document is brought to their attention this is an intrusion in their activities. Rhodes and Maes [25] have observed that, for CAR, *precision is generally more important than recall*, a key observation that influences many of the ideas presented in this paper. This is especially true when the retrieval was not explicitly asked for by the user, i.e. in the proactive case. Thus, *we believe that improving precision is an increasing need for applications that deliver information ubiquitously.*

There is a further factor influencing the need for precision: there is often physically a narrower bandwidth in communicating with the user than would be the case with a conventional desktop computer. Current applications use PDAs with tiny screens or perhaps audio delivery. Even if in future products there is a larger display area, e.g. information projected onto a wall or into the user's enhanced reality, the user's attention will not be solely focussed on that information. This contrasts with a static user whose attention is often concentrated on the screen of their desktop computer.

This observation will be less true of interactive retrieval initiated directly by the user, though even here, if the user is constrained by a small screen, they cannot easily browse through reams of information. Overall a useful maxim for the design of CAR applications is: *assume each retrieval brought to the attention of the user is an intrusion; therefore try only to deliver items that are both relevant and cannot be handled automatically via some form of agent acting on the user's behalf*. Even if the information is relevant, it still needs to be presented to the user in an appropriate manner; we return to this topic in Section 6.

### 2.3 Context Attached to Documents

In an ideal world each document would be marked up with the context associated with it in a way that can be readily matched against the fields of the user's context. Thus if the user's context contains location, temperature and time fields, these same fields should be attached to each document to be matched, giving the location, temperature and time associated with the information in the document. For example the document for each tourist site should give its location, the temperatures at which a visit would be suitable, and the opening times. (If a field is not relevant it can be given the infinite value 'ANY'.) In reality, however, an application will need to work with documents prepared by outside organisations, and with legacy documents: in neither case is it likely that context is attached to documents in the way the application wants.

In such cases the application needs to derive the context from the mark-up or the content of a document. If available, HTML markup of the document can be used to identify context information, e.g. the address of a restaurant; as more documents become available marked up using XML more meaningful context is likely to be easily identifiable. In addition *information extraction* techniques developed for natural language processing can be employed to identify entities and their relationships [11]. Currently information extraction techniques are often restricted to narrow domains, but development of more general and robust methods is an active area of research.

Particularly when derived automatically, the association of a document with particular context values need not be treated as exact. The association could be treated as a likelihood (e.g. from examining the text of a document, it might be inferred that there is a 60% likelihood that the document relates to a location in Canterbury, England); this likelihood can be incorporated in the overall matching score between the search topic and the document.

## 3 Information Selection and Delivery

Information can be selected for delivery in a number of ways. The traditional paradigms are information retrieval and information filtering. Those documents matching a search query are delivered to the user, who then inspects the documents to extract information relevant to their need. These can be extended to incorporate recommendations based on profiles derived from the user or groups of equivalent users. Such recommendation algorithms are an additional source of search topic contents, but we do not pursue them further here. The following sections define the basic features of IR and IF systems and highlight relevant differences.

### 3.1 Information Retrieval

Most people are now familiar with the use of IR systems in the form of web search engines. The retrieval engine responds to a search request by returning a set of potentially relevant documents to the user.

Each matching score gives a weighting of how well the document matches the query. In CAR systems these matching scores are even more important: as well as being useful for ranking, they can be used in deciding whether to deliver any documents at all. For example a proactive system may decide that, since the best matching document still has a rather low score, it is not sensible to distract the recipient with it; thus nothing is delivered. While highly desirable, it is important to note that thresholding criteria such as these are notoriously difficult to determine in current IR and IF systems [26].

Most real-world applications involve retrieval from a huge number of possible documents, and unless some optimisations are made, there will be performance problems. Thus a lot of research has been devoted to such optimisations: the basic strategy is normally to take those parts of the data which are relatively static and to preprocess these parts and place them into carefully designed data structures, so that the retrieval engine can do its matching more quickly.

### 3.2 Information Filtering

In IF systems the user's interests are again represented by queries which describe their information need, but here these queries are often referred to as *profiles*. IF systems are aimed at relatively stable, long-term information needs, although IF systems usually allow these interests to be modified gradually over time as conditions, goals and knowledge change. In this environment, rather than actively searching collections, users are often more passive, waiting for individual documents to be brought proactively to their attention. IF systems typically apply the same text preprocessing strategies as IR systems to improve efficiency and reliability of matching between profiles and documents. Documents with matching scores exceeding a threshold are passed to the user [26].

IF systems raise their own issues of efficiency. Many systems support thousands of simultaneous users, and thus a single document is compared in parallel to a potentially very large number of profiles. Efficiency can be achieved here by using an inverted file of the search profiles [1].

### 3.3 Document Structure

For both IR and IF, the simplest approach to matching of queries/profiles with documents is to treat the whole document as a single object. However, when the document is divided into distinct fields, it is straightforward to take these into account in the matching process if desired. The use of document structure is particularly pertinent to CAR where, as outlined earlier, the document structure will usually be extended to include its associated context fields, e.g. location, time, etc.

### 3.4 Developments in Context

The more an application knows about the user's context, the more likely it is that it can deliver documents the user wants. Increasingly sensors are available to record the user's physical context, and the values from these sensors translate into a rich array of contextual fields. The area of wearable computing accentuates this trend.

Contextual fields such as location are becoming straightforward to infer by extracting sensor information and relating this to sources such as maps. Much more difficult is the aggregation of sensor information to determine the user's current activity [10]. This is important to ensure that the presentation of information is appropriate to the user's activity and consistent with the sensed event. For example, the media of presentation can be varied according to the opportunity to devote attention to it. Aside from the theoretical development of such multi-sensor inference technologies, various practical requirements need to be taken into account. For example, context must be computed quickly and the hardware should be generally be cheap.

A current example of work in this area is the use of accelerometers to determine user activity [23]. A single accelerometer mounted in a user's jacket is used with a neural network classifier to determine whether the wearer is sitting, standing, walking or running. This information could obviously be combined with other information from the environment, e.g. information about whether the wearer is talking or listening could be detected by a microphone mounted in their jacket. Such information can help determine whether the user should be interrupted at this time.

In addition applications need to cater for a rich variety of information sources. There are doubts [25] whether a retrieval engine should aim to cater for several *simultaneous* information sources, but there is no doubt that a retrieval engine should be able to use different sources at different times. A resource selection method might be used to determine the most appropriate source given the current context. Moreover there is the problem that the content of some sources, such as one concerned with traffic information, will be highly dynamic, and thus not amenable to those retrieval optimisations, highlighted in Section 3.1, that depend on static information that can be pre-processed in advance.

A consequence of all this richness is that retrieval will become slower. Even with current context-aware applications there are countless stories along the lines of 'the system delivered just the information the user needed about available trams, but by the time the information was delivered the best tram had departed'. IR has been immensely successful in delivering information fast, even when searching over a billion documents; CAR technologies, which involves different parameters, must be developed that do the same. This observation leads us to consider what additional features of context might be exploited to help achieve this.

## 4 The Nature of Context Change

For CAR we have the challenge of selecting documents with high precision in a short time from document collections that may be dynamic. To meet this challenge we need to find some retrieval advantages that apply to CAR. We believe that the most important advantage is that the current context is often changing gradually and semi-predictably. Based on this we have developed two tools [3], the Context-aware Cache, which we describe in Section 8, and the Context-of-Interest, which we describe below.

In order to capture change we have introduced a structure called a *Context Diary*, which is described in detail in [3]; the Context Diary maintains a record of previous contexts and expected future contexts.

### 4.1 The Context-of-Interest

In many CAR applications, particularly mobile ones, the user may often not in fact be interested in information relating to their current context: instead they are likely to be interested in a context 'just ahead'. This is an example of what we refer to from now on as the *context-of-interest.* For example the context-of-interest of a traveller or tourist might be set with the aim that they retrieve information just before they need it. The Context Diary can be used, together with the current context, to predict the context-of-interest. This predicted context-of-interest is then passed to the retrieval system in place of the true current context, with the aim of retrieving documents that are more relevant to the user's needs at the time of delivery.

As an example of how a field within the context-of-interest may be used, a Location field may be set to a point (or, more likely, a range of values) ahead of the user's current location, taking into account their direction and rate[2] of travel, since the user is more likely to be interested in sites ahead of rather than behind them. Of course prediction can be wrong. However our experiments so far indicate that modest predictions — taking a small rather than a large leap into the future — are, on balance, winners, at least in tourist applications.

Finally the context-of-interest can be used to improve the setting of the current context. Some sensors give occasional totally wrong values, and others periodically fail to work (e.g. GPS in a tunnel). Prediction can be used for checking and to smooth over difficult periods; the result should be an improvement in the relevance of delivered documents, and an elimination of some irrelevant ones.

## 5 Personalisation

So far we have only considered the incorporation of environmental context into the retrieval process. Another important context component is the individual user for whom the retrieval operation is taking place. Ubiquitous applications do not *per se* distinguish between different users. Nevertheless modelling the interests of individual users or user groups (e.g. proximity services [27] and collaborative filtering [12]) is clearly an important issue for effective CAR.

[2] And potentially expected rate of progress, e.g. as predicted by monitoring traffic reports.

There are currently many projects underway exploring methods of *personalisation* for information seeking. The general starting point is to represent the interests of a user by means of one or more keyword profiles expressing aspects of the user's interests. In a very simple approach incoming information is compared with the profiles and passed to the user if there is a sufficiently high match. This scenario is very similar to standard IF, where the profiles represent topics of interest to the user. It can be combined with IR by either using the information from the profiles to enrich a query prior to retrieval by adding additional words to it, and/or after retrieval by using the profiles to rescore retrieved documents in an attempt to bring those most of interest to the individual user nearer to the top of the list.

An important issue in personalisation for retrieval is how the interests of the user are captured. Various options are available; important ones are as follows: the user selects from a number of preset topics, the user enters sets of keywords which they believe represent topics that they are interested in, or the personalisation system monitors the user's behaviour and learns profiles from this. In all cases once initial profiles have been acquired there is scope for the system to continue to adapt over time as more information is gathered. For example, user web and email habits can be monitored and used to identify interests, which can then be clustered into themes [6] [8]. Ongoing monitoring enables changes of interest to be detected and profiles to be changed.

A practical example of ubiquitous computing which could be extended for personalised CAR is the "Shopping Jacket" [22]. Sensors in the wearer's jacket and in nearby shops communicate to inform the user if desired items are available in the shops, possibly to compare prices between competing retailers, and to attempt to entice the wearer into the shop with special offers on products that relate to their interests. This scenario could easily be extended to incorporate CAR for more general applications, matching user interest profiles against information relevant to the user's current context and activity.

## 6 Application of Agents for CAR

In Section 2 we noted that proactive CAR will often be an intrusion into the user's activities and in the light of this suggested that this intrusion should only take place if really necessary. One aspect of this is to seek to filter out unimportant or unreliable information, a topic to which we return in the next section; another complementary approach is to make use of automation to process information of behalf of the user so that action is taken without requiring their attention.

Applications able to fulfil this requirement are often referred to as *agents*. An agent application of this type would monitor the context and available information, compute the importance of the information and interpret it, determine the action to be taken and then act. The initial stages of this are proactive retrieval, but interpretation requires the application of *information extraction* techniques to identify data within the retrieved documents, and then some form of "intelligence" to determine the appropriate action. The issue of how an agent acquires *competence* is a critical question for agent applications: typical methods used are rule-based systems, expert systems or some form of machine learning [16].

The delegation of responsibility for the interpretation and use of information in a ubiquitous computing environment raises another key topic in agent technology, that of *trust*. The user must trust that the agent will act correctly when receiving new information. The user's trust of the agent will relate to several aspects; the user will tend to trust the agent if they were personally involved in determining its competence, e.g. by designing its rule set, or by observation if the agent performs well over time.

Note here that we are not suggesting that agents should sit between the information and the user, but rather that they should adopt the role of the *personal assistant* working alongside the user, as advocated in [16]. Thus the user is free to act themself if they wish, rather than let the agent take charge. They may, for example, ask their networked information appliance "what information is currently available which might be of interest to me?".

Agents, like human assistants, are typically good at repetitive tasks (a) for which they are directly programmed by the user, (b) which they have observed the user perform many times or (c) which other agents have informed them of.

### 6.1 Deciding When to Interrupt the User

Another important area for which agents can be used in ubiquitous CAR is the decision of when to deliver information to the user. Agents can monitor the user's current activities from the available sensor information, as outlined in Section 2, and use this to determine when and how to deliver information to the user. For example, if the user is currently moving forward while driving in heavy traffic it is probably not a good time to inform them of a suggested route half a mile further up the road: better to wait until later when the traffic has cleared. If the user is then stationary it may be quickest to show the user the suggested route on a map. If the user is moving forward freely when the information needs to be delivered, showing the information on a map would be very dangerous and delivery via an audio description would be better. The concept of the context-of-interest introduced earlier is important here as well. By monitoring current activities, e.g. cars ahead of the driver moving forward, the agent can anticipate that the user, though currently stationary, will soon be occupied driving and that now is not a good time to begin delivery of information.

The area of driver disturbance from mobile phones and satellite navigation systems is currently contentious, with their role in causing accidents a high public concern. One aim of ubiquitous computing should be reduce such accidents, not to be the cause of them! Thus, we can also look to the agent in its role of personal assistant to decide whether the user needs to know this information at all. If the user typically ignores advice from the source of the information, or if a route to be suggested is one that the agent has observed the user to take before anyway, then the agent can decide not to pass it on to the user.

Again like a human assistant, the agent should only take action if it has a sufficient degree of confidence that this action will be in accordance with the user's wishes. Thus it should pass to the user information that is either novel, since the agent will not know what to do, or of unknown relevance or authority, since the agent will be uncertain as to whether an action is appropriate. (You would not want an agent to book a hotel for you, if (a) that hotel only had partial relevance to your needs, or (b) the only authority was that the hotel was recommended by its owners.)

### 6.2 Information Transformation

Another important role for agents in retrieval and management of information for ubiquitous computing is that of *intermediaries* that transform information as it flows from one computer to another in order to tailor it for the current circumstances [17]. For example, documents may be summarised to aid efficient delivery of key information (this may include the summarisation of multiple documents into a single summary); alternatively a document may be transformed from one media to another, e.g. a text document may be passed through a speech synthesizer for audio delivery, or a spoken document may be analysed by a recogniser to provide a text rendering.

Further examples include the possibilities of translating documents, useful for example for someone travelling in a foreign country, or of annotating documents for the individual user (an alternative form of information personalisation) [25].

### 6.3 Information Authority

One important factor in delegating responsibility for document and information management to an agent is the authority, importance and reliability of an information source. Some aspects of this can be inferred from the personalisation methods discussed in Section 5. For example, a source frequently acted on by a user can be regarded as reliable and important. An additional aspect in determining the action to be taken over some information relates to the *authority* conferred on it by its relationship to other available information [14].

Traditionally IR and IF techniques have focussed on the matching of document contents with query/profile expressions, and have handled documents as disjoint entities. More recently attention has been devoted to the topic of document authority. One application for these techniques is illustrated by the many broad topic queries, e.g. `find me details on PDAs`, entered by users of web search engines; often for these queries many thousands of pages can be identified as potentially relevant. Information retrieval algorithms give methods to rank the documents, but for short broad queries the resulting ranking owes more to chance word distribution statistics than meaningful selection of relevant documents. Much more useful, once a set of potentially relevant documents has been identified, is to make use of the authority assigned to each of them by users. One expression of document importance is the *conferred authority* expressed by latent human judgement of relevance indicated by the number of other documents which have hypertext links to it. This has the added feature that important documents related to a search topic can be identified even if they don't contain the search words, e.g. for the query `search engines` many relevant pages would fail to match (since search engine homepages rarely contain the words `search` and `engines`), but many documents which do discuss search engines explicitly will point to them. We can confer further authority by identifying groups of documents that point to the same pages, indicating that not only is the authoritative page pointed to by many other pages, but further that they are pointed to by pages that point to many relevant documents.

The concept of document authority may be particularly relevant for CAR where we are aiming for high precision. The authority of the document, computed either directly or inferred from its source, could be used to determine whether this information is likely to be important and/or reliable; this could be further incorporated with the matching score threshold in determining whether the document should be delivered. One example of an agent system is *Amalthaea* [19]. Amalthaea is a system for personalised news delivery; this uses a multi-level ecosystem with learning via genetic algorithms both to personalise profiles to user interests, and also to learn about the importance and reliability of individual sources in providing information of interest to a specific user.

## 7 Practical Investigation and Evaluation of CAR

In the last part of the paper we focus on our own practical work. In the first part of this paper, we explored a number of issues for information management in ubiquitous computing environments. In order to develop and test these ideas we have developed an experimental CAR platform. In a separate paper [3] we describe the software architecture of our CAR engine.

### 7.1 Experimental Testbed

Our experimental testbed consists of a set of exchangeable components. One central component is a retrieval engine: ideally this could be an existing IR engine, but, for our own purposes, given our need for experimentation, we have built our own engine. Other components are largely concerned with massaging data, for example (a) a context-of-interest component that pretends the user's context is slightly ahead of its true value on the basis of future prediction or (b) a component to adjust scores or change the weights of fields to factor in past history. This architecture of exchangeable components, possibly including conventional IR engines, is also an aid to covering the spectrum between the two retrieval extremes of (a) conventional, entirely user driven, IR and (b) proactive retrieval with no direct user control.

### 7.2 Evaluation

An important aspect of research in information retrieval is the evaluation of precision and recall for the technology under investigation. We believe that this is no less true of CAR. We need to have a way of measuring precision, both to show whether our ideas really can deliver an improvement over existing IR and IF methods, and to help in tuning the various algorithms that the system uses.

One of the features of developing applications for ubiquitous computing environments is the difficulty of testing them prior to the widespread practical realisation of systems. An interesting approach to addressing this problem is introduced in [5]. Taking a graphical virtual world games engine, the authors explore ways in which a games engine can be used to simulate a user's activity in a virtual world into which monitoring of context and indeed elements of ubiquitous computing could be introduced. Such a scenario would enable the application of many different ubiquitous technologies to be explored in simulation, with the most promising investigated experimentally in the future.

### 7.3 Matching Algorithms

A focus of our recent work has been to find good matching algorithms that improve precision. As we have said, at the heart of a CAR system is the basic retrieval engine. At the heart of the retrieval engine is a matching algorithm that will take the user's context together with a potential document to be retrieved, and come up with a score on how well the two match. A context will usually consist of a set of different fields, and these fields will generally cover different data types. (As we have also said, if the user makes an explicit retrieval request, i.e. by specifying some search terms, it is convenient to treat this as a field of the context, along with the rest; in an extreme case it would be the only field.)

The matching algorithm typically works by computing a score for each field in turn, and then aggregating the results. (A more sophisticated algorithm might work with the fields in combination, but we will keep to the simple case here.) Matching of textual fields is a topic much studied in IR, and well-developed algorithms are available. Matching of numeric fields, such as locations given by pairs of co-ordinates, is a much more open field. An interesting approach to this problem is taken for a topic tracking application in [18]. In this work named locations are matched using a hierarchical taxonomy, for example Paris is in France, and this relationship is captured in the taxonomy. Temporal data is transformed into a standard form and overlap between topic and document recognized and rewarded. We have done some experiments following our earlier ideas in [13] and the 'fuzzy matching' techniques used by Rhodes [24], with a focus on delivering high precision, and have made the following, albeit rather limited, conclusions:

– In practice, fields are often ranges rather than single values. This applies particularly to contexts attached to documents. For example a document about a town might apply to the area covered by that town, i.e. a range of locations, and a document about frost precautions might apply to all temperatures below 4 degrees Centigrade (here we have a potentially infinite range, as there is no lower limit specified). Ranges may also be used in the user's context, e.g. when the user is known to be somewhere within a given room, or when the user has specifically requested a range as their sphere of interest. (Ranges can also be used to represent uncertainty due to inaccurate sensors, but we prefer this uncertainty to be encompassed in the matching algorithms.)

– An ideal would be generic algorithms, one for each data type, that applied to all fields of that data type. Thus one generic algorithm would cover one-dimensional numeric fields. Such generic algorithms are indeed useful as a default, but they need to be overridden in many individual cases. For example a generic algorithm, when matching a point against a range, might give the highest score if the point were in the middle of a range, with lower scores towards the edges of the range. If, however, a field represents opening-hours of a building (a range of times), and this is matched against the current time (a point in time), the match should give a high score if the current time was near the start of the opening-hours, a less high score for the middle, and a low score if near the end.
– If two ranges are matched, the score should be higher if the ranges are roughly the same size. Thus delivering a document about the county of Devon is especially appropriate if the user has requested information for an area of about the same size – and of course if the area overlaps with Devon too.
– There must be smooth behaviour as a range gets smaller and smaller and eventually becomes a point, or when a range becomes wider and wider until it becomes infinite.
– Finally there is the negative conclusion that we have been unable to find good algorithms to aggregate individual field scores to get an overall score. Using arithmetic or geometric means has severe flaws, but more elaborate algorithms often have equally severe, though perhaps less obvious, flaws.

## 8 Performance and Ubiquitous Availability

Three performance challenges for CAR are: (1) delivering information of high precision; (2) delivering it fast; and (3) delivering it ubiquitously, even when the mobile user is periodically disconnected. We have discussed all these above, but will now concentrate on giving some detailed suggestions for (2) and (3).

Speed is an especial challenge. Many context-aware applications try to give the user the illusion of continuous retrieval, e.g. as a user moves round an exhibition, information on their screen continually changes to reflect nearby stands, nearby potential contacts, etc. Solutions to the speed problem have tended to assume that the content of the information source is static, and we do that here. Indeed the case where all data is dynamic has been called 'the grand challenge' [20]. The possibilities for dynamic information repositories represented by networked mobile devices are described in [27], and handling such environments must be a long term research goal for CAR.

In traditional IR the approach to improving speed has been to build, from the content of the information source, surrogate structures that are such faster to search. Building these structures takes time, but if the information is relatively static it is time well spent.

In CAR we have proposed a surrogate structure called the *context-aware cache* [2, 3]. The context-aware cache tries to capture the documents the user is likely to need in future contexts that they are about to enter. In its current simple form it works as follows:

- The application sets a time span during which it thinks the cache will be useful; for our current data set of assuming tourist travelling in Devon, we set this to 20 minutes.
- On the basis of history and predicted future events (e.g. diary appointments in the immediate future) the application predicts what ranges each contextual field will cover in the next 20 minutes. Thus the predicted range for a location field might cover a range of a mile round the user's current point; if the user had been travelling in a fairly constant direction, this range might be biased towards locations ahead rather than locations behind.
- The application then does a retrieval where, within the user's context, the value of each field is replaced by its predicted range. The results of this retrieval are then treated as a cache. Generally the cache will be hugely smaller than the original source. (This retrieval may add a weighting factor based on past history, e.g. documents that have previously been retrieved by similar users within this context get added weight.)
- The cache is now used as the information source, rather than the original, retrieval should now be much faster.
- If the user's context strays out of its predicted range, the cache will become invalid. It will then be necessary to replace the cache or, in favourable circumstances, incrementally update it. If, contrary to our previous assumption, the content of the information source *is* dynamic the cache would need to be incrementally updated when changes occur. Strategies for incremental updating of the cache and, very importantly if rapid retrieval is paramount, for the updating of retrieval data structures, will need to be the subject of detailed further analysis and investigation.

Predicted contexts can sometimes be used independently of caches to improve performance [2]. For example if retrieval typically takes 10 seconds, then a retrieval request can use the user's predicted context in 10 seconds time.

The context-aware cache has some similarities to location-aware hoarding mechanisms [15], but the latter are concerned with explicit requests for documents, whereas our caches are concerned with anticipating the documents that are retrieved via a future query issued by the user — i.e. the user does not know the explicit documents they want. Our caches are designed to handle contexts that are much richer than just location, though location-aware hoarding mechanisms could be adapted to cover richer contexts too. Some initial results of our experiments with their use are found in [4].

We now move briefly on to issue (3): ubiquitous availability even when periodically disconnected. Caches stored on the user's personal device are, of course, a prime method of dealing with disconnected operation. The context-aware caches that we have proposed serve this purpose well. Their application ranges from short-term disconnectivity to cases in fieldwork [21] where the user is disconnected throughout a whole day, just docking with a base-station night and morning. Our implementation of context-aware cache algorithms is at an early stage, but our experiences thus far have proved useful in developing their specifications, and the issues which must be addressed in achieving these.

## 9 Summary

Context-aware retrieval needs to bring together a number of disparate technologies. In the case of the underlying technologies of IR and IF, CAR requires a new approach, taking features from each. Performance and method of delivery are crucial issues, and agent technology offers a means of tackling these. We have also discussed related issues in personalisation and information authority for CAR ubiquitous environments.

Towards the end of the paper, we discussed our own work. This work is built on what we hope is an apposite combination of existing IR and IF, extended to include new techniques to meet the challenges and opportunities of ubiquitous context-aware environments. These new methods include the *context-of-interest*, which seeks to deliver relevant information when the user needs it, and the *context-aware cache*. The latter addresses issues of the potentially very high number of search queries associated with rapidly and continually changing context, and also addresses problems arising from discontinuities in network connectivity.

## References


1. Belkin, N.J. and Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? *Communication of the ACM*, 35:29-38, 1992.
2. Brown, P.J. and Jones, G.J.F.: Context-aware retrieval: exploring a new environment for information retrieval and information filtering. *Personal and Ubiquitous Computing*, 5(4):253-263, 2001.
3. Brown, P.J. and Jones, G.J.F.: Exploiting contextual change in context-aware retrieval. In *Proceedings of the 17th ACM Symposium on Applied Computing (SAC 2002)*, Madrid, pp. 650-656, 2002.
4. Brown, P.J. and Jones, G.J.F.: Evaluation of straying outside a context-aware cache. http://www.dcs.ex.ec.uk/~pjbrown/cgi-bin/stray_evaluation, 2002.
5. Bylund, M and Espinoza, F.: Testing and demonstrating context-aware services with Quake III Arena. *Communications of the ACM*, 45(1):46-48, 2002.
6. Crabtree, B. and Soltsiak, S.: Identifying and tracking changing interests. In *Proceedings of a IJCAI'97 Workshop on AI in Digital Libraries*, 1997.
7. Dourish, P., Bellotti, V., Mackay, W. and Chao-Ying Ma.: Information and context: lessons from a study of two shared information systems. In *Proceedings COOCS'93*, San Jose, pp. 42-51, Nov. 1993.

8. Fisher, M.J. and Everson R.M.: Representing interests as a hyperlinked document collection. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM'03)*, New Orleans, 2003.
9. Ford, D.A., Ruvolo, J., Edlund, S., Myllymaki, J., Kaufman, J., Jackson J. and Gerlach, M.: Tempus Fugit: a system for making semantic connections. In *Proceedings of Tenth International Conference on Information and Knowledge Management (CIKM 2001*, Atlanta, pp. 520-522, 2001.
10. Gellersen, H.W., Schmidt, A. and Beigl, M.: Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications* 7:341-351, 2002.
11. Grishman, R.: Information Extraction: Techniques and Challenges. In *Information Extraction*, Pazienze, M. T. ed., pp. 10-27, 1997.
12. Gurrin, C., Smeaton, A.F., Hyowon Lee, McDonald, K., Murphy, N., O'Connor, N. and Marlow, S.: Mobile access to Fishlár-News Archive. In this volume.
13. Jones., G.J.F. and Brown, P.J.: Information access for context-aware applications. In *Proceedings of ACM SIGIR 2000*, Athens, pp. 382-4, July 2000.
14. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604-632, 1999.
15. Kubach, U. and Rothermel, K.: Exploiting Location Information for Infostation-Based Hoarding. In *Proceedings of the Seventh ACM SIGMOBILE Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, pp. 15-27, July 2001.
16. Maes, P.: Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7), 1994.
17. Maglio, P. and Barrett, R.: Intermediaries Personalize Information Streams. *Communications of the ACM*, 43(8):96-101, 2000.
18. Makkonen J., Ahonen-Myka H., and Salmenkivi, M.: Topic Detection and Tracking with Spatio-Temporal Evidence. In *Proceedings of the 25th European Conference on IR Research (ECIR 2003)*, Pisa, Italy, pp. 251-265, 2003.
19. Moukos, A. and Maes, P.: Amalthaea: An evolving multi-agent information and discovery system for the WWW. *Autonomous Agents and Multi-Agent Systems*, 1(1):59-88, 1998.
20. Oard, D.W. and Marchionini, G.: A conceptual framework for text filtering. Report EE-TR-96-25, Univ. of Maryland, 1996.
21. Pascoe, J., Morse, D.R. and Ryan, N.S.: Developing personal technology for the field. *Personal Technologies* 2(1):28-36, 1998.
22. Randall, C. and Muller, H.: The Shopping Jacket: wearable computing for the consumer. *Personal Technologies* 4(4): 241-244, September 2000.
23. Randall, C. and Muller, H.: Context awareness by analysing accelerometer data. In *Proceedings of the Fourth International symposium on Wearable Computers*, pp. 175-176, October 2000.
24. Rhodes, B.J.: The wearable remembrance agent: a system for augmented memory. *Personal Technologies*, 1(1):218-224, 1997.
25. Rhodes, B.J. and Maes, P.: Just-in-time information retrieval agents. *IBM Systems Journal*, 39(4):685-704, 2000.
26. Robertson, S., and Walker, S.: Threshold setting in adaptive filtering. *Journal of Documentation*, 6(3):312-331, 2000.
27. Touzet, D., Weis, F. and Banâtre, M.: PERSEND: enabling continuous queries in proximate environments. In this volume.