EXPLOITING CONTEXTUAL CHANGE IN CONTEXT-AWARE RETRIEVAL

Peter Brown and Gareth Jones

Dept of Computer Science
Univ. of Exeter, UK

Context-aware retrieval (CAR)

A new application area for retrieval. Some examples: **Proactive tourist guides; context detected by sensors; triggering condition (query) supplied by information provider. Interactive tourist guides, where query is enhanced by user's context. Proactively adding links to document on user's screen; context is computational one. Exploitation of personal preferences, history. Office applications geared to role and activity. Reminders, memory aids. Commercial services: 'Tell me the nearest Macdonalds'. CAR concepts**

Needs expertise from retrieval and mobile computing and ... . User's current context is typically a collection of fields, covering various data types (text, location, time, image, etc.). CAR is related to IF/IR, but not the same. Continuous retrieval as context changes. Retrieval may be intrusive: high

precision is vital; may decide to deliver nothing at all. The richer the context, the better the retrieval. CAR matching

Retrieval involves a match of user's context, field by field, against context associated with each document. Field values may be ranges rather than points. Three research areas for improving precision: v algorithms for matching individual fields, especially non-text fields. v algorithms for weighting fields. v algorithms for accumulating overall scores. ... BUT evaluation is hard. Context change

CAR is hard relative to IF/IR; one compensating advantage may be that context changes gradually and semi-predictably. To record context change we build a *Context Diary***; it covers past and future. We use the Context Diary to predict the** *Context-of-interest***, which is just 'ahead' of the current context. This can be used to improve speed and**

precision. Context-aware caching: v predict the union of a user's current contexts over the next, say, 15 minutes. This union consists of context fields whose values are ranges. v retrieve the documents that best match the union, and use these as a cache. v cache will improve speed, and cater for disconnected operation. v works best when small changes in context lead to small changes in what is retrieved. Using context change to improve precision

Possible approaches: a changing field should have more weight than a currently static one. ***rate*** **of change is significant. relevance feedback can also affect weights. Current experimental system**

Retrieval engine written in Java; allows considerable tailoring, e.g. supplying a new matching algorithm for Location. Use of pipeline architecture, e.g.:

v calculate user’s context ⇒ v massage user’s context ⇒ v proactive retrieval ⇒ v interactive retrieval to factor in user preferences ⇒ v massaging final scores, using the context diary. Summary

High-performance CAR is a need for many future mobile applications. CAR needs high precision, fast speed (even with fully dynamic data). Analysis of context change provides a means of achieving these needs. Techniques include: v recording a context diary. v predicting the context-of-interest. v using a context-aware cache.