# Mining Large-Scale Knowledge Sources for Case Adaptation Knowledge

David Leake and Jay Powell

Computer Science Department, Indiana University, Lindley Hall 215
150 S. Woodlawn Avenue, Bloomington, IN 47405, U.S.A.
{leake, jhpowell}@cs.indiana.edu

**Abstract.** Making case adaptation practical is a longstanding challenge for case-based reasoning. One of the impediments to widespread use of automated case adaptation is the adaptation knowledge bottleneck: the adaptation process may require extensive domain knowledge, which may be difficult or expensive for system developers to provide. This paper advances a new approach to addressing this problem, proposing that systems mine their adaptation knowledge as needed from pre-existing large-scale knowledge sources available on the World Wide Web. The paper begins by discussing the case adaptation problem, opportunities for adaptation knowledge mining, and issues for applying the approach. It then presents an initial illustration of the method in a case study of the testbed system WebAdapt. WebAdapt applies the approach in the travel planning domain, using OpenCyc, Wikipedia, and the Geonames GIS database as knowledge sources for generating substitutions. Experimental results suggest the promise of the approach, especially when information from multiple sources is combined.

## 1 Introduction

Case adaptation is a classic challenge for case-based reasoning (CBR). One of the impediments to endowing CBR systems with automated case adaptation is that adaptation often requires substantial domain knowledge, which may be difficult to capture. Knowledge-based adaptation methods have been widely explored in research systems (see [1] for a survey), but their application is limited by practical concerns: the difficulty and expense of hand-coding knowledge, as well as difficulties in anticipating how cases may be used, may make it infeasible to encode adequate adaptation knowledge in advance. Machine learning methods have been explored to extract adaptation rules from the case base for future use (e.g., [2]), and to capture adaptation experiences (e.g., [3]). Nevertheless, it is still common in CBR applications to follow the advice which Barletta [4], Kolodner [5], and others advanced in the 1990's: leave adaptation to the user.

This paper proposes addressing the knowledge capture problem for case adaptation by exploiting the large-scale, publicly-available knowledge sources now available on the Web. The goal is to develop largely domain-independent methods for "just in time" mining of domain-specific information as needed for specific adaptations, to give CBR systems robust adaptation capabilities without requiring the specific details of the adaptation domain to be precoded by CBR system-builders. To our knowledge, this is the first application of Web mining to the adaptation task.

R.O. Weber and M.M. Richter (Eds.): ICCBR 2007, LNAI 4626, pp. 209–223, 2007.

The success of Web mining to support case adaptation will depend both on the form and the coverage of the Web knowledge sources and on the ability of the CBR system to extract relevant knowledge. Not all Web sources are currently suitable for simple knowledge extraction, but we hypothesize that enough useful sources exist to make Web mining a valuable approach for enabling CBR systems to adapt a greatly increased set of problems. As initial support for this hypothesis, the paper presents a feasibility study exploring the use of three sources to support case adaptation in the travel planning domain: OpenCyc, a formalized knowledge-base of general-purpose knowledge [6], Wikipedia, a natural language encyclopedia including some structured information [7], and the Geonames GIS database, a database of site types and locations [8].

The paper begins by identifying key issues for harnessing general-purpose knowledge sources for case adaptation. It then explores some of these issues through a study introducing the WebAdapt system, a program which mines Web information to propose adaptations to tourists' sight-seeing itineraries. WebAdapt applies largely domain-independent strategies to extract domain-specific information as it is needed for substitution adaptation; this makes the methods applicable to adapt tourism itineraries without manual knowledge capture of details of local attractions or pre-processing of knowledge sources. The paper presents encouraging results from initial system tests, and closes by discussing prospects, limitations, and open questions for developing general frameworks for mining the Web to support case adaptation.

## 2 Framing the Problem

Any general approach to the adaptation problem must be based on a characterization of the task and required knowledge. For many tasks, case adaptation can be characterized in terms of two parts: (1) a small set of abstract structural transformations (e.g., [9]), and (2) memory search strategies for finding the information needed to apply those transformations, by substituting appropriate components into the case structure. For example, this view can be applied to tasks such as case-base planning (e.g., when a new ingredient must be substituted into a recipe [10]) and case-based explanation (e.g., when a plausible alternative cause must be substituted to replace a previous cause which does not apply to the current situation [11]).

The adaptation system's assessment of suitability for a substitution must be based on a characterization of the role to be filled. This can be described by a set of constraints to be satisfied, which can guide a search or "knowledge planning" through the internal knowledge of the CBR system, following the basic model of Kass's Tweaker [12] and Leake, Kinley and Wilson's DIAL [3]. Such a process assumes that the system will be endowed with sufficient background knowledge to address any adaptation problem it may encounter, forcing the system developer to confront the knowledge acquisition bottleneck. However, if the system could effectively mine pre-existing external knowledge sources, the burden on the system builder might be significantly reduced. This paper explores the use of knowledge mined from large-scale sources both (1) to hypothesize constraints which a replacement element must satisfy, and (2) to find replacement elements satisfying such constraints.

The proposed approach is consistent with recent general observations, by Hendler [13] and others, that the World Wide Web may provide the infrastructure to break the knowledge acquisition bottleneck. As ontologies become increasingly prevalent, carefully-crafted knowledge covering task domains of interest will become more widely available, perhaps significantly alleviating the burden of capturing formalized case adaptation knowledge. However, because the availability of formalized knowledge is currently outstripped by that of more informal, human-centered knowledge sources built collaboratively by individuals for human use, the aim of the WebAdapt project is to explore how case adaptation can benefit from both types of sources, with special focus on informal sources with fairly constrained forms, such as Wikipedia. Recent work explores mining Wikipedia for natural language processing tasks such as computing semantic relatedness [14] and augmenting text categorization algorithms [15]. To our knowledge, the proposed approach is the first effort to harness such sources to support CBR.

*Central Issues:* Given the differences in knowledge coverage by different Web sources and the different adaptation needs for different domains, the practicality of a Web mining approach will vary with the domain, task, and candidate knowledge sources. For any domain, tasks, and knowledge sources, applying the approach will depend on answering the following questions:

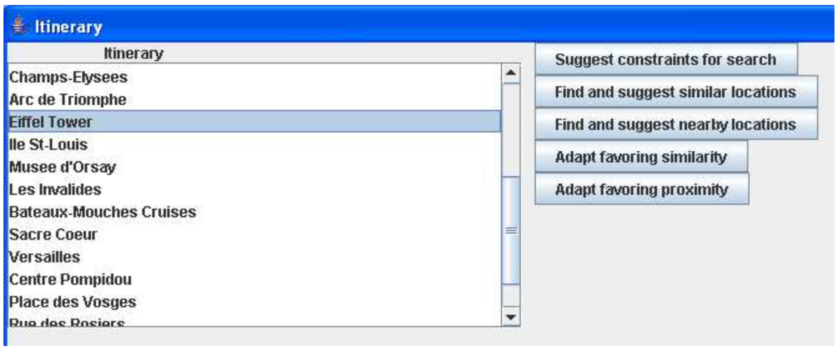
1. Which knowledge sources (or combinations of sources) should be exploited?
2. Which strategies should be used to determine constraints for knowledge search?
3. Which strategies should be used to mine each source for information satisfying the constraints?
4. For the task, knowledge sources, and generated constraints,
   - Are the knowledge source coverage and search strategies sufficient to find suitable information?
   - Is the mining process sufficiently efficient to make its application practical?

## 3 The WebAdapt System

Our testbed system, WebAdapt, explores the use of large-scale knowledge sources for identifying substitution adaptations using three knowledge sources which exemplify some major dimensions in the space of Web knowledge sources:

- OpenCyc, the open source version of the Cyc general-purpose knowledge base of formalized knowledge. OpenCyc contains hundreds of thousands of terms, and millions of assertions intended to represent consensus commonsense knowledge.
- The English version of Wikipedia, a collaboratively-written encyclopedia in natural language, including over 1.6 million articles.
- The Geonames GIS database, a database of over 8 million geographical names with associated features, including location information.

WebAdapt's task domain is tourism, an area which has been explored in a number of previous CBR projects (e.g., to support a community through sharing tour cases [16]). This is a domain for which generalized coverage of possible destinations and


Itinerary
Itinerary
Suggest constraints for search
Find and suggest similar locations
Find and suggest nearby locations
Adapt favoring similarity
Adapt favoring proximity

Champs-Elysees
Arc de Triomphe
Eiffel Tower
Ile St-Louis
Musee d'Orsay
Les Invalides
Bateaux-Mouches Cruises
Sacre Coeur
Versailles
Centre Pompidou
Place des Vosges
Rue des Rosiers

**Fig. 1.** Top level screen for WebAdapt

adaptations would require staggering amounts of precoded knowledge (e.g., Wikitravel includes guides to over 14,000 destinations, for each of which a traveler might consider multiple itineraries and substitutions).

WebAdapt's task is to aid user adaptation of plan itineraries, or to automatically adapt them. Its focus on the generation of substitutions for existing itinerary steps (e.g., if a user has already seen one of the proposed sights or prefers to see a different sight). The system's case-base contains itineraries drawn from the Frommer's travel guide to Paris [17], but because it processes itineraries in textual form and mines the Web for all additional knowledge needed, it could be applied to any domain for which sufficient Web coverage exists.

WebAdapt's user begins by selecting an itinerary case (e.g., "Best of Paris in 3 Days"). The system then presents the case's steps to the user. For steps which the user wishes to change, WebAdapt supports two adaptation modes, one in which the user interacts with the system to select constraints and alternatives from candidates proposed by the system, and another in which WebAdapt itself performs all adaptations.

Figure 1 shows the interface allowing a user to choose between interactively adapting an itinerary or having WebAdapt adapt it automatically. The steps in the itinerary are displayed on the left side of the screen, with user options on the right. For interactive adaptation, users can select the "Select constraints for search" button for WebAdapt to display a list of attributes hypothesized to be relevant for the selected item (e.g., reflecting that Notre Dame is a church and an example of Gothic architecture). The user can then select a constraint of interest, for WebAdapt to expand the constraint and display a hierarchy of retrieved sub-constraints and items. To have WebAdapt generate the suggestions autonomously, the user selects either "Find and Suggest Similar Locations" or "Find and Suggest Nearby Locations" for WebAdapt to generate constraints, use them to guide search, and display a ranked set of top suggestions to the user. The user may also select to have one of the top suggestions substituted into the itinerary.

## 3.1 Extracting Role-Filler Constraints

To generate constraints to guide the search for substitutions, WebAdapt applies a set of domain-independent search strategies. Each strategy calls on source-specific procedures to handle the format of information in each knowledge source (e.g., seeking Wikipedia category information at the bottom of the page, or examining all collections for an item in OpenCyc). However, it relies almost entirely on general-purpose approaches for its information search process in order to test the power of general approaches. We note that for any specific domain, it would be possible for a knowledge engineer to select a set of constraints *a priori*, or to hand-code task- and domain-specific constraint generation strategies, e.g., based on user profiles reflecting individual user interests, or to define domain-specific search procedures. This effort would be expected to increase the accuracy of results, and could be applied in conjunction with the more general methods investigated here.

Constraints are generated in a two-step process, first retrieving an initial set of "seed constraints," then performing refinement/filtering on the initial retrieval results. Given an item in an itinerary, the initial search strategies query the knowledge sources for information about the item. Currently, items in WebAdapt's cases undergo no re-coding from the text in Frommer's; each step is described by a set of keywords, which are used to query each knowledge source for an associated entry (e.g., to find substitutions for the Louvre, the system queries the sources for entries concerning the keyword "Louvre"). Entries for proper names or general categories may be found simply by matching keywords with the node labels in a formal representation (e.g., for OpenCyc), or by matching terms in the titles of associated Web pages (e.g., for Wikipedia). Keeping itineraries in textual form was chosen to test the ability of the system to function with minimal knowledge engineering.

Depending on the domain and representation scheme, additional pre-processing could be needed before constraints can be hypothesized. For textual role-fillers, NLP methods could be applied, using the context of the rest of the case to facilitate disambiguation; if items are encoded in a formal knowledge representation, role-based strategies could be used to select terms to generate Web queries [18].

*Generating seed constraints:* Each of the three Web knowledge sources includes an explicitly-defined abstraction hierarchy. If the knowledge source contains an entry (or entries) corresponding to the item to replace, WebAdapt considers each node in the source's abstraction hierarchy as a potential constraint. For the knowledge sources used in our tests, the seed constraints are:

1. Wikipedia - Categories under which an item falls
2. OpenCyc - General collections under which an item falls, regardless of any microtheories
3. Geonames GIS database - The feature class of an item

WebAdapt's core search process involves ascending and descending each source's abstraction hierarchy. The abstraction hierarchy is descended whenever a node is reached that can be expanded. In OpenCyc this will occur when a node corresponding to a collection is found, and in Wikipedia when a category node is found. The Geonames GIS abstraction hierarchy does not contain any child nodes which can be expanded (that is, the only node associated with each Geonames object is the item's feature class).

*Expanding seed constraints to search for potential fillers:* Once WebAdapt has generated a candidate set of role-filler constraints, it performs a breadth-first search for related information, expanding each constraint by searching its parents and recursively searching its children to find associated items. Individual items are listed in the order in which they are encountered. Search depth is limited, with no branch being searched to a depth greater than five nodes from a seed constraint; a limit is needed because the Wikipedia category structure occasionally contains cycles. Experience with these knowledge sources suggests that most relevant information is typically discovered at a depth of one or two nodes away from a seed constraint.

In our tests, WebAdapt finds one to eleven seed constraints when searching for role-filler information. Wikipedia routinely returns the largest number of role-filler constraints, averaging six per query. OpenCyc returns an average of three hypotheses, with a maximum of six, while the GIS Geonames database only returns one hypothesis per query. The retrieval of objects associated with a typical constraint for any of these knowledge sources usually returns approximately twenty objects. A retrieved object from OpenCyc is the name of an OpenCyc constant, an object from Wikipedia is a string of text describing a page, while an object from the Geonames database is the name of an item, it's feature class, and geographical coordinates.

*Filtering result sets:* For both Wikipedia and Geonames, WebAdapt relies on source-specific methods for filtering the retrieved items. For Wikipedia, WebAdapt's simple task-specific heuristic is to search for the name of the tour location in the first paragraph of the Wikipedia entry for each object in the initial retrieval set. If the name of the tour location is not found within the first paragraph, the item is rejected.

Geonames objects are filtered by location, requiring that objects must be within a three mile radius of the tour location (in this case, Paris, assuming that the user is primarily interested in items near the heart of a city). Tests of other radii between three and seven miles produced a fairly uniform set of results. No source-specific filtering is used for OpenCyc, as its knowledge is carefully encoded and any extra information that is associated with each object tends to be concise and unambiguous (e.g., the comment or 'pretty string' for the OpenCyc object Louvre-Museum is "the Louvre").

WebAdapt's other refinement processes include taking the intersection of the result sets returned by each knowledge source, or giving objects that simultaneously satisfy multiple constraints more weight. The intersection of two or more knowledge sources is taken by comparing each individual result set one element at a time, and retaining items that have the same name, or are judged equivalent based on another Web mining strategy, described in the following subsection. When searching for objects that simultaneously satisfy multiple constraints within one knowledge source, the number of constraints an object satisfies is stored. Objects satisfying the most constraints are displayed first.

## 3.2 Finding Replacement Elements That Satisfy Multiple Constraints

The initial search process often generates numerous candidate objects, many of which are irrelevant. To help refine this set, WebAdapt first attempts to resolve references described in different ways by different sources, then applies its role-filler constraint extraction process to each item, and then uses the constraints to estimate relevance, either automatically or based on user feedback about constraint importance.

*Resolving naming inconsistencies:* A system mining natural language must contend with the inconsistencies that presents. A particular sight may be referred to in different ways, but, ideally, the system should resolve these references to determine a unique set. In the spirit of exploiting large scale knowledge sources, WebAdapt uses Google to resolve naming conflicts. For each name mined from the text, WebAdapt queries Google and records the first URL returned. If the names from two knowledge sources are mapped to the same hyperlink address, they are considered to be equivalent items. For the examples in the experiments, this method had approximately 90% accuracy.

*Determining object relevance:* WebAdapt's automated strategy weights retrieved objects by the number of constraints that they share with the seed constraints for the initial role to be filled. Objects that share the greatest number of constraints are ranked highest; objects that share only one constraint with the seed constraint are ranked lowest.

In WebAdapt's interactive method, users define their preferred constraint types as they modify an itinerary. As a user modifies an itinerary, WebAdapt stores the final result of each modification and the system asks the user to select the constraint which was most important in his or her choice. For example, a user searching for a replacement for Notre Dame de Paris may prefer to look for other churches. If WebAdapt is using Wikipedia to suggest alternatives, the Wikipedia category "Churches in Paris" is most salient. WebAdapt keeps statistics on each constraint choice, building up an implicit model of user interests. The weighting of items mined in the future is adjusted, based on the number of high-ranked constraints they share with stored user preferences.

*Filtering redundant candidates:* Depending on the specificity of the constraints generated for a substitution, some of the retrieved items may duplicate objects already in the itinerary. To guard against suggestions for redundant items, WebAdapt compares each candidate result with every itinerary item already in the case, using the previously-described method for resolving naming ambiguities. Items from the result set that are determined to refer to the same sight as an element in the current case are discarded from the candidate set.

## 3.3 Generality of the Strategies

The WebAdapt system's constraint generation process is domain-independent; the system finds abstractions of the current role-filler. To find those abstractions, the system uses general strategies such as ascending and descending the abstraction hierarchy defined by a knowledge source. Likewise, the processes for generating and expanding seed constraints are independent of both the domain and knowledge source.

Because different sources organize knowledge differently, the procedures for tasks such as extracting abstractions are specific to the particular knowledge source, although not to any particular domain (e.g., the procedures for searching through the category hierarchy of Wikipedia are applicable to any domain, not only the tourism domain).

The filtering process involves some procedures specific to the type of knowledge involved and characteristics of the knowledge sources. For example, Geonames results are filtered by location, which applies only to spatial information; Wikipedia articles are filtered based on the assumption that the first paragraph of a Wikipedia article contains an overview of the most salient information on the article's topic. The process for resolving naming ambiguities applies to domains where items can be described by proper names which are easily queried.

## 4 Evaluation

Our evaluation studies the characteristics of alternative knowledge sources, explores the ability of the constraint generation and search processes to retrieve and rank relevant candidate substitutions, and tests the potential for the mining of multiple sources to improve performance. Specifically, the experiments were designed to provide information on the following questions:

1. How do the selected general purpose knowledge sources compare for suggesting useful adaptations?
2. Are the mining and ranking strategies successful at extracting information in the sources?
3. Can performance be improved by selecting items that simultaneously satisfy constraints from multiple sources?

For question 3, we considered both the combination of Web sources and a simple approach to refining use of Web sources by learning from interactive adaptations.

*Experimental Design:* The experiments tested adaptation suggestions for two itineraries taken from Frommer's Paris Travel Guide [17]:

1. The Best of Paris in 3 Days, a tour of 25 sights
2. Walking Tours: The Marias, Montmartre, The Literary and Artistic Left Bank, and The Latin Quarter, a tour of 60 sights

The 'Best of Paris' tour itineraries contain well-known Parisian tourist attractions; the 'Walking Tours' are small themed tours of less renowned attractions such as the living quarters of famous artists or small shops and cafes. One itinerary of each type was used in order to compare the system's ability to acquire adaptation information for (1) widely-recognized items likely to be contained in several knowledge sources, and (2) obscure items less likely to appear. Adaptation was performed on each of the steps in the itineraries, for a total of 85 adaptation problems.

Experiments were conducted using OpenCyc 1.0, the Geonames database updated as of January 7, 2007, and using Wikipedia pages last updated on February 10, 2007.

*Performance Measures:* Our experiments address two types of questions. First, they explore the domain coverage when WebAdapt's methods are applied to each particular source, i.e., the pool of items WebAdapt retrieves, given the constraints it generates from the problem and the source.

Second, they explore WebAdapt's ability to suggest the right items from that pool to present to the user, measured by standard precision and recall measures, applied to the sets of suggestions appearing in WebAdapt's top $k$ suggestions, for $k \in \{1, 3, 5, 10\}$. As a coarse-grained impartial measure for which items in the pool are relevant, we consider an item relevant if it is mentioned in Frommer's Paris Travel Guide.

**Table 1.** Number of items returned from each knowledge source for various items to replace

| Object | | Wikipedia | OpenCyc | Geonames GIS |
|---|---|---|---|---|
| Notre Dame de Paris | Items returned | 61 | 163 | 41 |
| | Items within Paris | 58 | 19 | 41 |
| | In Frommer's Guide | 40 | 16 | 20 |
| Arc de Triomphe | Items returned | 60 | 94 | 41 |
| | Items within Paris | 58 | 10 | 41 |
| | In Frommer's Guide | 32 | 10 | 20 |
| St Germain Des Pres | Items returned | 137 | 179 | 0 |
| | Items within Paris | 136 | 19 | 0 |
| | In Frommer's Guide | 17 | 16 | 0 |
| Place Des Vosges | Items returned | 63 | 0 | 5 |
| | Items within Paris | 62 | 0 | 5 |
| | In Frommer's Guide | 14 | 0 | 4 |
| No. 20 Rue Jacob | Items returned | 26 | 0 | 0 |
| | Items within Paris | 0 | 0 | 0 |
| | In Frommer's Guide | 0 | 0 | 0 |

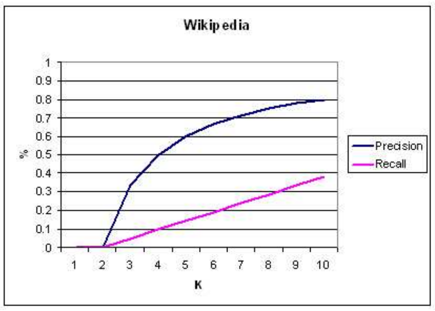
*Comparing knowledge sources:* In our studies, Wikipedia provided substantially more coverage than OpenCyc or Geonames, as illustrated in Table 1 for five sample sights in an itinerary. The table lists the total number of items returned for each sight, how many of those were actually within Paris, and how many were listed in Frommer's. Notre Dame de Paris had a high percentage of relevant items returned from each knowledge source, while the more obscure No. 20 Rue Jacob had no relevant items returned from any knowledge source.

*Case study of substitutions for the Eiffel Tower:* To illustrate observed performance in more depth, we report results for the adaptation task of finding a substitution for the Eiffel Tower, which was chosen as a representative example illustrating variations in knowledge coverage and the quality of generated constraints. Table 2 shows the constraints generated using each knowledge source. Mining Wikipedia resulted in a rich set of constraints, ranging from describing the Eiffel Tower as a tourist attraction in Paris, to identifying the architectural time period under which the Eiffel Tower was constructed. Several of these constraints relate specifically to Paris or France itself, such as identifying the Eiffel Tower as a skyscraper in Paris, or as a tower or landmark in France. The results from Wikipedia also include some constraints, such as "Articles with unsourced statements" and "Eponymous places," which are irrelevant to identifying alternative sights.

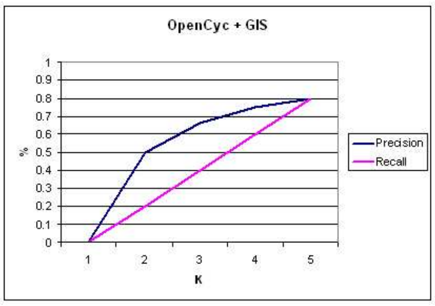
OpenCyc provided fewer constraints than Wikipedia, but those returned tended to be more focused than the Wikipedia constraints. The OpenCyc and Wikipedia constraints agreed on several key features of the Eiffel Tower, such as being a tourist attraction, a landmark, and a tower. The Geonames GIS database allows for only one constraint per item, tending to describe a fairly broad range of items, as illustrated in Table 2. This makes it less useful as a constraint source, except for providing the ability to restrict items geographically.

**Table 2.** Hypothesized constraints for substitutions for the Eiffel Tower

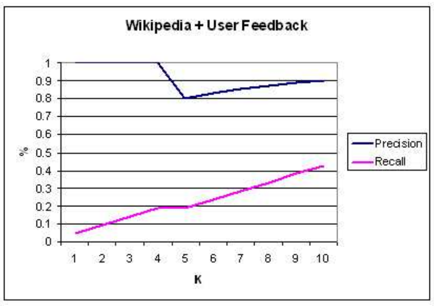
| Wikipedia | OpenCyc | Geonames GIS |
|---|---|---|
| Articles with unsourced statements | Landmark | Spot, building, farm, etc. |
| Visitor attractions in Paris | Tourist Attraction | Paris coordinates: $48.87^o$ N, $2.33^o$ E |
| 1889 architecture | Tower | |
| Former world's tallest buildings | | |
| Historic civil engineering landmarks | | |
| Landmarks in France | | |
| Michelin Guide | | |
| Skyscrapers in Paris | | |
| Towers in France | | |
| Eponymous places | | |



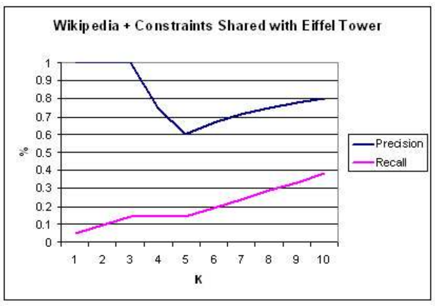
(a) Baseline Wikipedia precision



(b) OpenCyc ∩ GIS precision/recall



(c) Wikipedia + user feedback + compiled adaptation knowledge precision/recall



(d) Wikipedia + shared constraint information precision/recall

**Fig. 2.**

When WebAdapt used the generated constraints on substitutions for the Eiffel Tower, Wikipedia returned 32 items, 29 of which were within Paris, and 21 in Frommer's guide, compared to 170, 18, and 14 for OpenCyc and 41, 41, and 20 for Geonames (Geonames' high percentage within Paris is expected, due to WebAdapt's filtering for items within three miles of the center of Paris).

Of the items returned that were actually located in Paris, Wikipedia and OpenCyc contained a significant percentage of items identified by Frommer's as tourist attractions (approximately 75% in both instances). However, only 50% of the Geonames results were tourist related. Geonames contains a wealth of information about items within Paris, ranging from information on office buildings and communes, to the locations of several farms.

Figure 2(a) illustrates the precision and recall measures for the Wikipedia baseline example for the first ten items returned. The first two items displayed to the user when using Wikipedia alone are "2005 civil unrest in France" and "Visitor attractions in Paris." The object "2005 civil unrest in France" corresponds to the constraint "Articles with unsourced statements," while "Visitor attractions in Paris" is a general article elaborating on the phrase naming the constraint. Items two through ten each were displayed because they corresponded to the constraint "Visitor attractions in Paris," where each of the last eight items in this set appears in Frommer's Paris travel guide. The precision/recall measurements for OpenCyc were zero for the first ten items returned, while the precision measure for the Geonames GIS database was 60%, as six of the top ten items returned from the GIS database were found within the Frommer's tour guide. Recall at k = 10 for the GIS database was 49%.

*Combining results from multiple sources:* To test the effects of combining knowledge sources, after evaluating results using each knowledge source separately we compared the results of using intersections of the results from each of the individual sources, as well as of combining constraint information from the three sources and of using preferences learned from simulated user interactions. Conditions were:

1. Wikipedia ∩ OpenCyc ∩ Geonames GIS database (GIS)
2. Wikipedia ∩ OpenCyc
3. Wikipedia ∩ GIS
4. OpenCyc ∩ GIS
5. Wikipedia + Shared Constraint Information
6. Wikipedia + User feedback + Learned preference information

Conditions 5 and 6 favor items that simultaneously satisfy multiple constraints. In 5, a purely automated approach weights suggested items by the number of hypothesized constraints that intersect with the original item to be adapted. In 6, the system observes those constraints that the user of a system seems to prefer, and places greater weight on those items whose hypothesized constraints intersect with the user's preferences.

System results for intersections may sometimes have gaps, due to failures of WebAdapt's simple approach to resolving inconsistent naming. For the test example, the system-generated intersection of Wikipedia and OpenCyc produced five items, all of which were in the Frommer's Paris guide: the Arc de Triomphe, the Opera Garnier

Ceiling Painting, the Louvre, the Musée D'Orsay, and the Sacré Coeur. However, the true intersection also includes three more items, Notre Dame de Paris, the Pantheon, and Montmartre.

The intersection of Wikipedia and the Geonames database produced five results, each of which were found in the Frommer's travel guide. Each of these items simultaneously satisfied the constraints "Visitor attractions in Paris" and "Spot, building, farm." Figure 2(b) displays the results of the intersection of OpenCyc and the GIS database. The first item returned was the only irrelevant item, "Pantheon Rome." This item was incorrectly suggested due to the simple name merging strategy, which incorrectly mapped items corresponding to the Pantheon in Paris to the Pantheon in Rome.

Figure 2(c) and 2(d) illustrate precision and recall for a strategy that preferred items from one knowledge source that simultaneously satisfied multiple constraints. For the strategy that relied upon user feedback and user compiled adaptation knowledge, the only invalid suggestion was the object "Visitor attractions in Paris." Overall, filtering by constrains from several knowledge sources noticeably improved the precision and recall values compared to baseline strategies.

## 5 Related Work

Previous CBR research on acquiring adaption knowledge has explored a number of methods. Wilke et al. [19] propose approaches for refining adaptation knowledge using knowledge already contained in the CBR system; methods for mining adaptation knowledge from pre-existing cases include work by Hanney and Keane [2], Craw, Jarmulak and Rowe [20], and Patterson, Rooney, and Galushka [21]. Such approaches have proven valuable, but what they can glean is limited by the knowledge already contained within the case-base itself. Research on areas such as mining cases from databases has begun to address the question of leveraging external knowledge (e.g., the case mining of Yang and Cheng [22]), as has the use of case mining to extract cases from problem data, followed by mining adaptation rules from those cases, as by Patterson and Annad [23]. The WebAdapt approach contrasts in not acquiring its knowledge from an existing case-base or set of problems; instead, it mines large-scale, freely available knowledge sources developed for other purposes; the project's goal is to develop a flexible framework for enabling CBR systems to draw on multiple pre-existing knowledge sources. In addition, the goal of WebAdapt is not to generate new general adaptation rules, but to find the specific information needed to apply general adaptations.

The WebAdapt model of searching for adaptation knowledge is similar to that of the Leake, Kinley and Wilson's DIAL system, which also framed adaptations in terms of transformations and memory search. However, DIAL searched only internal structured knowledge, and relied on explicitly pre-specified role-filler constraints to provide goals for its search process. The WebAdapt system generates its own constraints and does a "just in time" search of external sources, taking a lazy approach to gathering adaptation knowledge and guided top-down by current needs, rather than bottom-up by the available data. McSherry's on-demand adaptation using adaptation triples [24] is in a similar spirit.

WebAdapt 's capture of user-selected adaptations to favor in the future is in the spirit of DIAL's manual adapter and of other case-based approaches to adaptation [25]; it also relates to Aquin et. al's CABAMAKA system, which combines case base mining with expert guidance [26].

Because WebAdapt searches sources which may vary greatly in its capabilities and internal structure, one of the challenges for WebAdapt is developing and managing the right search strategies. We anticipate that methods from information extraction and textual CBR research [27] to prove relevant.

## 6 Future Issues and Outlook

Current WebAdapt research has suggested many open issues for future study. The suitability of the Web mining approach for a particular domain will depend on its needs and sufficiently rich knowledge sources, but too much knowledge could impair the efficiency of the system, as the number of alternatives grows (currently, WebAdapt's constraints for the Louvre yield roughly 100 OpenCyc items, while its constraints for Place de la Concorde return 8,800). If the number of returned items becomes unmanageable, methods will be needed to increase search efficiency or generate more selective constraints (e.g., in the travel domain, to use Geonames information to only return alternatives within a small distance from the original tour location). Currently, WebAdapt mines a prespecified set of sources; another relevant question is how to automatically select a few on-point sources to which to dispatch queries for a particular problem (e.g., [28,29]).

Also, more sophisticated methods are needed for generating constraints in the context of the case as a whole, and for filtering and ranking search results. As mentioned previously, we are already exploring capturing user preferences from prior problems, for case-based reuse of successful adaptations across different users in similar contexts. Finally, human subjects studies will be needed to assess the overall success of the methods.

Despite the continuing challenges, we consider the initial results highly encouraging. WebAdapt can often propose good adaptations relying only on its cases, knowledge sources external to the system, and a few simple rules for mining them. Thus for domains with a good fit to available Web sources, the Web mining approach may be a promising avenue for helping to alleviate the knowledge bottleneck for case adaptation. The approach has great promise for adapting itineraries: With no additional knowledge capture effort, such a method could provide suggestions for an enormous number of destinations, limited only by the knowledge encoded in sources such as Wikipedia.

Procedures to learn constraints from prior adaptations to focus future choices, and to learn specific adaptations favored by users, could help to focus system results. We expect such procedures to be an important focus for the next phase of the project. In addition, based on the lessons from WebAdapt, we aim to develop a framework of general domain-independent methods for extracting constraints, defining knowledge source characteristics, and managing search, to facilitate application of the methods to new tasks and knowledge sources.

## Acknowledgment

We thank Thomas Reichherzer, Matthew Whitehead and the anonymous reviewers for helpful comments on a draft of this paper.

## References

1. Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision, and retention in CBR. Knowledge Engineering Review 20(3) (2005)
2. Hanney, K., Keane, M.: The adaptation knowledge bottleneck: How to ease it by learning from cases. In: Proceedings of the Second International Conference on Case-Based Reasoning, Springer, Heidelberg (1997)
3. Leake, D., Kinley, A., Wilson, D.: Learning to improve case adaptation by introspective reasoning and CBR. In: Proceedings of the First International Conference on Case-Based Reasoning, pp. 229–240. Springer, Heidelberg (1995)
4. Barletta, R.: Building real-world cbr applications: A tutorial. Presented at the Second European Workshop on Case-Based Reasoning (1994)
5. Kolodner, J.: Improving human decision making through case-based decision aiding. AI Magazine 12(2), 52–68 (Summer 1991)
6. Cycorp: OpenCyc. Accessed (February 17, 2007) (2007), at `http://www.opencyc.org/`
7. Wikimedia Foundation: Wikipedia. Accessed (February 17, 2007) (2007), at `http://www.wikipedia.org`
8. Geonames: Geonames. Accessed (February 17, 2007) (2007), at `http://www.geonames.org`
9. Carbonell, J.: Learning by analogy: Formulating and generalizing plans from past experience. In: Michalski, R., Carbonell, J., Mitchell, T. (eds.) Machine Learning: An Artificial Intelligence Approach. Tioga, Cambridge, MA, pp. 137–162 (1983)
10. Hammond, K.: Case-Based Planning: Viewing Planning as a Memory Task. Academic Press, San Diego (1989)
11. Kass, A., Leake, D.: Case-based reasoning applied to constructing explanations. In: Kolodner, J. (ed.) Proceedings of the DARPA Case-Based Reasoning Workshop, pp. 190–208. Morgan Kaufmann, San Mateo, CA (1988)
12. Kass, A.: Tweaker: Adapting old explanations to new situations. In: Schank, R., Riesbeck, C., Kass, A. (eds.) Inside Case-Based Explanation, pp. 263–295. Lawrence Erlbaum, Mahwah (1994)
13. Hendler, J.: Knowledge is power: A view from the semantic web. AI Magazine 26(4), 76–84 (2005)
14. Strube, M., Ponzetto, S.: Wikirelate! computing semantic relatedness using wikipedia. In: Proceedings of the Twenty-first National Conference on Artificial Intelligence, AAAI Press, Stanford (2006)
15. Gabrilovich, E., Markovitch, S.: Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In: Proceedings of the Twenty-first National Conference on Artificial Intelligence, AAAI Press, Stanford (2006)
16. Blanzieri, E., Ebranati, A.: Supporting touristic culture via cbr. In: Blanzieri, E., Portinale, L. (eds.) Proceedings of the Fifth European Workshop on Case-Based Reasoning, pp. 358–369. Springer, Heidelberg (2000)
17. Frommer's: Frommer's Paris 2006. Frommer's (2006)

18. Leake, D., Birnbaum, L., Hammond, K., Marlow, C., Yang, H.: Integrating diverse information resources in a case-based design environment. Engineering Applications of Artificial Intelligence 12(6), 705–716 (1999)
19. Wilke, W., Vollrath, I., Althoff, K.D., Bergmann, R.: A framework for learning adaptation knowedge based on knowledge light approaches. In: Proceedings of the Fifth German Workshop on Case-Based Reasoning, pp. 235–242 (1997)
20. Craw, S., Jarmulak, J., Rowe, R.: Learning and applying case-based adaptation knowledge. In: Aha, D., Watson, I. (eds.) Proceedings of the Fourth International Conference on Case-Based Reasoning, pp. 131–145. Springer, Heidelberg (2001)
21. Patterson, D., Rooney, N., Galushka, M.: A regression based adaptation strategy for case-based reasoning. In: Proceedings of the Eighteenth Annual National Conference on Artificial Intelligence, pp. 87–92. AAAI Press, Stanford (2002)
22. Yang, Q., Cheng, S.: Case mining from large databases. In: Case-Based Reasoning Research and Development: Proceedings of the Fifth International Conference on Case-Based Reasoning, ICCBR-03, pp. 691–702. Springer, Heidelberg (2003)
23. Patterson, D., Anand, S., Dubitzky, W., Hughes, J.: Towards automated case knowledge discovery in the M$^2$ case-based reasoning system. In: Knowledge and Information Systems: An International Journal, pp. 61–82. Springer, Heidelberg (1999)
24. McSherry, D.: Demand-driven discovery of adaptation knowledge. In: Proceedings of the sixteenth International Joint Conference on Artificial Intelligence (IJCAI-01), pp. 222–227. Morgan Kaufmann, San Mateo (1999)
25. Sycara, K.: Using case-based reasoning for plan adaptation and repair. In: Kolodner, J. (ed.) Proceedings of the DARPA Case-Based Reasoning Workshop, pp. 425–434. Morgan Kaufmann, San Mateo, CA (1988)
26. d'Aquin, M., Badra, F., Lafrogne, S., Lieber, J., Napoli, A., Szathmary, L.: Case base mining for adaptation knowledge acquisition. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07), pp. 750–755. Morgan Kaufmann, San Mateo (2007)
27. Weber, R., Ashley, K., Brūninghaus, S.: Textual case-based reasoning. The Knowledge Engineering Review 20, 255–260 (2005)
28. Leake, D., Scherle, R.: Towards context-based search engine selection. In: Proceedings of the International Conference on Intelligent User Interfaces, pp. 109–112 (2001)
29. Leake, D., Sooriamurthi, R.: Case dispatching versus case-base merging: When MCBR matters. International Journal of Artificial Intelligence Tools 13(1), 237–254 (2004)