



NORTHWESTERN UNIVERSITY

Electrical Engineering and Computer Science Department

Technical Report
NWU-EECS-06-14
November 21st, 2006

Context Transformations for Just-in-time Retrieval: Adapting the Watson System to User Needs

Jay Budzik, Sanjay Sood, Kristian J. Hammond, and Larry Birnbaum

Abstract

The Watson system automatically retrieves useful information in the context of a document-manipulation task. Previously-reported laboratory studies show the basic Watson system can consistently retrieve useful documents. However, a number of deployments in real-world settings suggest improvements in retrieval performance can be made. We describe a novel set of techniques for automatically transforming representations of a user's context for the purpose of just-in-time information retrieval. The techniques are grounded in the experience gained during pilot deployments of the system. The methods described allow the system to identify salient properties of text the user is manipulating. This and additional representations of the user's context are then used to guide a set of reusable query formation strategies, allowing the system to deliberately retrieve information of known relevance in the context of specific classes of tasks.

Keywords: Just-in-time information retrieval, software agents, context, functional indexing

Context Transformations for Just-in-time Retrieval: Adapting the Watson System to User Needs

Jay Budzik

Center for Technology Commercialization
Northwestern University
1801 Maple Ave., Evanston, IL 60201 USA
+1 847 467 6158
budzik@devlab.northwestern.edu

Sanjay Sood, Larry Birnbaum, and Kristian J. Hammond

Intelligent Information Laboratory
Northwestern University
1890 Maple Ave., Evanston, IL 60201 USA
+1 847 491 3500
{sood, birnbaum, hammond}@infolab.northwestern.edu

ABSTRACT

The Watson system automatically retrieves useful information in the context of a document-manipulation task. Previously-reported laboratory studies show the basic Watson system can consistently retrieve useful documents. However, a number of deployments in real-world settings suggest improvements in retrieval performance can be made. We describe a novel set of techniques for automatically transforming representations of a user's context for the purpose of just-in-time information retrieval. The techniques are grounded in the experience gained during pilot deployments of the system. The methods described allow the system to identify salient properties of text the user is manipulating. This and additional representations of the user's context are then used to guide a set of reusable query formation strategies, allowing the system to deliberately retrieve information of known relevance in the context of specific classes of tasks.

Keywords

Just-in-time information retrieval, software agents, context, functional indexing

INTRODUCTION

Watson is an information access assistant that automatically retrieves useful information in the context of an ongoing task [3]. Watson integrates with existing applications in order to gain access to documents users are manipulating. The system analyzes the user's active document in the application in which it is being manipulated in order to compute a representation of her current work context. This representation is then used to form queries to multiple online information repositories. The system collects the results of the queries, sorts and filters them, and presents them to the user in a separate window. Essentially Watson provides users with an awareness of online resources that may be useful to her in the context of her current task.

In addition, Watson allows its users to engage in a context-

aware conversation about the information it accesses on behalf of the user. Users can select regions of text within the current document on which they would like the system to focus. This causes terms in the selected region to be promoted in importance in the lexical portion of the system's context representation. Users can also pose follow-up queries in the context of the document they are manipulating. The system uses its representation of the user's context to augment explicit queries, with the aim of reducing the effects of ambiguity in the short, natural language queries typical of most users of online information systems [16]¹.

Watson has been evaluated previously and has been shown to improve a user's ability to access relevant information when compared with their use of search engines alone [3]. Additional experiments show that users do find many of the documents retrieved by the system are *relevant*. Yet in the context of a specific task, users find fewer of the documents *useful* [1, 3], although more relevant than results retrieved using search engines alone.

In order to make just-in-time information systems truly useful, we need to reexamine the "similarity assumption" inherent in many of these systems' designs. That is, the goal of the system should not simply be to retrieve "more like this"² [1, 7, 9-11, 13-15]. Instead, the goal of the system should be to retrieve useful information, regardless of its resemblance to the user's current document. After all, the most similar document to the user's current document is an exact copy.

In light of this reasoning, and the results of several pilot studies of the Watson system, we propose techniques that bring modest amounts of knowledge to bear in order to (1)

¹ For example, out-of-context, it's impossible to determine whether the phrase, "oracle delphi," refers to a partnership between two software companies, or the shrine to Apollo in ancient Greece. However, when considered in the context of other words used in the same document, the reader (and systems) can easily disambiguate among potential meanings. For example, additional words like: "software product information technology," clearly indicate the first choice, while words like: "ancient greece apollo omphalos pythia," clearly indicate the second one. These additional words result in more precise queries because they eliminate the ambiguity inherent in shorter queries, more typical of human searching behavior.

² See <http://www.google.com/> for one example of sophisticated "more like this" functionality that goes beyond lexical similarity.

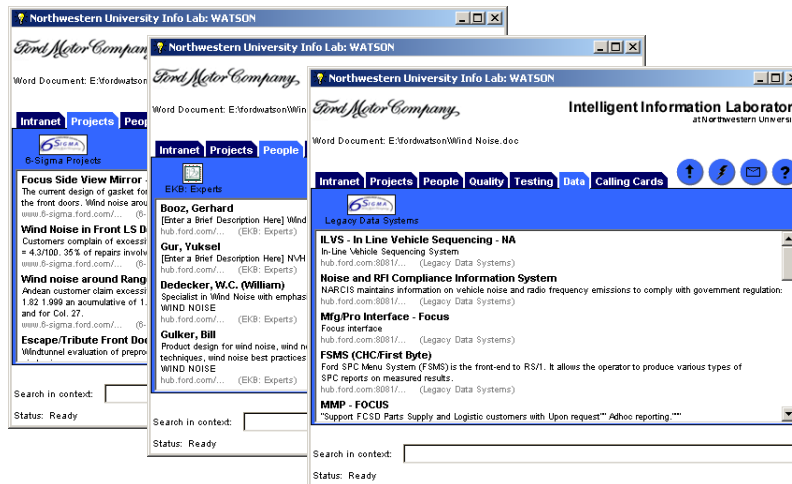


Figure 1: Screen shots of the prototype version of Watson developed for Ford Six Sigma Black Belts.

derive a representation of the user's context from her active document in the context of the application being used to manipulate it and (2) perform lexical transformations on this context representation in order to formulate queries to information systems aimed at retrieving not just similar documents, but documents that are relevant and useful in purposeful and interesting ways to the user engaged in a task.

The above techniques are grounded in experiences with Watson in several pilot deployments. These experiences provide a basis for the development of the techniques described above. A revised system architecture and implementation that flexibly supports this functionality is presented.

THE FORD MOTOR COMPANY SIX SIGMA PROTOTYPE

Ford Motor Company, a large automotive manufacturing company, employs the Six Sigma [12] process in order to improve the quality of their products and services. Six Sigma Black Belts are a highly-trained task force of quality assurance experts, that work like a consultancy within Ford. Black belts work on 6-10 week projects aimed at improving the quality of business and manufacturing processes throughout the organization. At Ford, Six Sigma projects typically include an intense data discovery phase in which the Black Belt attempts to uncover documents relevant to his project. Watson was seen as a technology that could help Six Sigma Black Belts uncover documents, data resources, and expertise relevant to their current project by linking relevant data repositories together in a single, context-aware interface.

Ford had recently completed an audit of their internal systems. Over 9,000 systems were identified in active use. Such systems contain the data Six Sigma Black Belts need in order to do their job. Yet some of the repositories in the inventory were only known to a handful of users and IT executives.

Black Belts use a project tracking database to track the progress of their projects, and to capture the results of engagements. The project database is intended to allow Black Belts to share their experiences so that common solutions can be reused, and past failures can be avoided.

Black Belts have a variety of IT resources available to them. Ford Motor Company hosts a searchable index of all internal Web sites available within the Ford firewall. Ford maintains and encourages the use of their Expertise Knowledge Base (EKB), which contains contact information for subject-matter experts (SMEs) within the company. Ford also has a database of quality assurance guidelines for each of its vehicle lines and for internal business processes. In addition, Black Belts have, and make extensive use of, Web access.

After an initial set of discussions with Six Sigma Black Belts in which we identified information resources both online and offline, we built a custom version of Watson for them in collaboration with Ford IT that worked within Ford's environment, and could automatically retrieve: (1) references to past Six Sigma projects documented in the Six Sigma project tracking database, (2) relevant information from Ford's intranet Web sites, (3) contact information for SMEs relevant in the context of a given document, (4) relevant information from external sites like JD Powers and Google, (5) potentially relevant databases audited during the inventory, and (6) it could retrieve pointers to relevant testing facilities using a database derived from interviews with relevant personnel.

Figure 1 contains screen shots of the prototype we developed for Ford. This version of the system included additional activity synchronization features, as in [4]. There were several lessons we learned as a result of developing the Ford prototype.

Issues uncovered by developing the prototype

First, the scale associated with the number of potential information sources challenged the basic Watson model. It

would not be feasible to have Watson search over 9,000 information sources simultaneously. Instead, we chose to have Watson search several key data sources, and then search a database containing descriptions of the other sources. This enabled Six Sigma Black Belts to determine if there was a more specific information asset relevant to their project and how to contact its owner, should it prove necessary to use³.

Second, many of the data sources Watson would be configured to search were structured databases, instead of information retrieval engines. The version of Watson we developed for Ford was heavily modified in order to retrieve information from such databases. Given a database, tools allowed the system to automatically build a set of named entity detectors based on lexical descriptions contained in database entries. When entities were detected, the system would automatically retrieve matching records from the database.

Third, differences in the use of language inside and outside of the company introduced complications when Watson went to retrieve documents from Internet information sources. For example, within Ford, products are typically referred to by the region in which they are distributed, in addition to their familiar product name, e.g., the “NA Focus,” or, “North American Focus,” instead of the “Ford Focus”. This led to the development of a translation table that is used to externalize the language used in internal documents before attempting to generate queries to retrieve information from public sites.

THE BIGCHALK PROTOTYPE

Bigchalk.com is a leading provider of online educational content. Used by thousands of schools nationwide, their Bigchalk Library product contains encyclopedia articles, definitions, timelines, and multimedia content covering a broad array of subject areas. Students can browse the Library by subject or search it from Bigchalk’s Web site. Teachers can access the same content, as well as custom lesson plans and descriptions of classroom activities.

Though Bigchalk has had major success selling its product to school districts, teachers and students alike do not use it as much as Bigchalk would like. Our thought was that Watson could help Bigchalk promote utilization of its Library product by making opportunities to apply Bigchalk content more visible. As a user works, Bigchalk content would automatically be retrieved and delivered using Watson.

A prototype version of Watson that searches Bigchalk Library was developed. Given that Bigchalk.com’s customers formed two groups—teachers and students—the prototype we developed reflected this division by allowing the user to switch between teacher and student modes. In student mode, Watson would retrieve media and articles

from Bigchalk Library, pictures from Google Images, news articles from CNN, and Web pages from Google. In teacher mode, the system would also search Bigchalk library for articles at higher reading levels, and lesson plans made available through Bigchalk’s Homework Central resource. In addition, it would search the Web for lesson plans posted by other teachers on Web sites. It would do this by automatically adding the terms “lesson plan” to the Google search query Watson would have normally executed. Thus for a document on any subject, a teacher could view a lesson plan on the same subject that would allow her to bring new and relevant learning experiences to the classroom.

As part of the prototyping process, a more detailed study of users and their computational environments was performed in order to determine how to best design the Watson system to fit the needs of students and educators. The initial discovery phase comprised several visits to two different school districts.

Our interviews and observations suggested typical high school students do not perform research while they are writing. This may be due to constraints introduced by their available computing environments (e.g., many of the students did not have Web access from home and had separated the two activities according to access) or instruction (e.g., students were instructed to perform research first, then write). A more structured interface that supports the task of writing a research paper might be more appropriate for certain students.

The interviews suggested that Watson would be more successfully used in a student environment if its use was scaffolded by supporting material, tools and processes. In the future, we plan to provide task-driven wizards that provide a structured environment in which tasks like paper-writing can be performed that are fully integrated with Watson. Such a structured environment will allow Watson to tailor its results to the various stages of the task; for example, given knowledge of the user’s task state, Watson could retrieve overviews in the beginning of the task, and more detailed references near the end.

THE MOTOROLA BUSINESS INTELLIGENCE PILOT

Motorola is a major cellular telephone, telecommunications and semiconductor company. Business intelligence is a company-wide effort aimed at maintaining an awareness of market demand for Motorola products and services, and analogous products and services offered by competitors. This awareness is intended to be integrated into decision making at all levels of the company.

Business intelligence (BI) researchers attempt to anticipate the release of potentially competitive or disruptive products by conducting informal meetings with employees of competing companies, attending trade meetings, monitoring postings to Web sites, and reading industry trade journals and research reports. Business Intelligence researchers can be deployed reactively or proactively. In

³ The automated strategy proposed by Leake, et al. [8] might be useful here, as well.

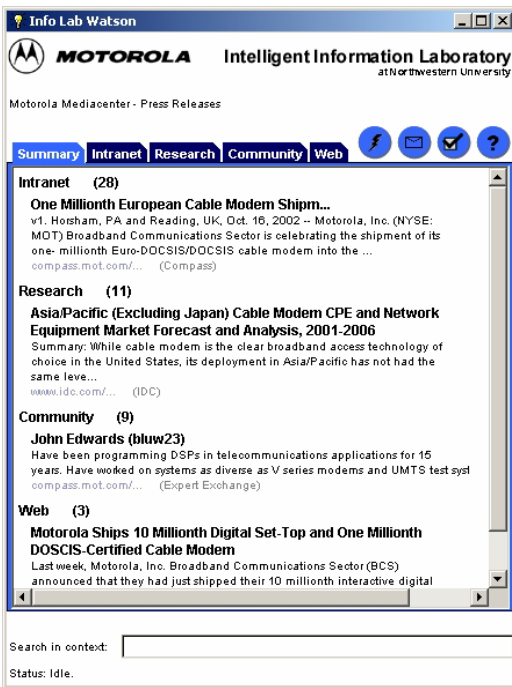


Figure 2: Version of Watson for Motorola business intelligence researchers.

practice, most of the research conducted by Corporate BI is reactive, as a result of a request from an executive.

Business intelligence researchers have access to information sources ranging from collections of industry research and analysis prepared by firms like Gartner Research, to documents contained in internal knowledge management systems, to postings on Web sites and bulletin boards, to access to a large collection of contacts within Motorola and its competitors.

Although Watson could be used in a number of ways by the BI community (as a research tool, and as way to coordinate BI activity across sectors), and its customers (as a way of disseminating BI research), we focused on providing a tool for business intelligence researchers to support the research process.

In order to produce usable software for Motorola BI researchers, we adopted a design and customization process that attempted to uncover the situations in which Watson could add the most value by providing the most assistance to the BI community. In the initial phase of the pilot, we conducted interviews of BI researchers in order to uncover the description of their work process provided above, as well as to determine what information sources were most frequently used by researchers and for what purpose.

The version of Watson currently in use by business intelligence researchers at Motorola (see Figure 2) includes interfaces to Adobe Acrobat (and Reader), Microsoft Outlook, PowerPoint, Word, and Internet Explorer. It searches three internal Motorola knowledge management

databases: Compass, Compass Expert Exchange, and Compass Internal Web Sites. In addition, it searches the analyst sites Gartner, and IDC Research, as well as the Internet search engines Google and AltaVista. Thus four functionally different kinds of information sources are made available in Watson's context-aware framework: (1) internal sources containing documents posted by Motorola employees in internal knowledge management systems, (2) research reports from external subscription sources used by business intelligence researchers, (3) experts registered with Motorola's internal subject matter expert database, and (4) Web sites. This allows BI researchers to identify internal information resources, and experts relevant to their current research topic; it allows them to understand the marketplace from a broader perspective by viewing relevant analyst reports; it additionally allows them to determine what people are saying about their products on the Web, in general. The system also includes a preferences interface that allows users to turn information sources on and off at will, as well as update passwords used to access restricted content.

Watson was formally introduced to the Business Intelligence community as part of a company-wide business intelligence conference. Two dozen members of the broader Motorola business intelligence community attended.

Motorola Pilot Results

Two months after the conference in which Watson was presented, 92 users had registered and downloaded Watson. Most of the Watson users at Motorola heard about Watson by word of mouth, not by direct communication from us. Over two thirds of the Watson users at Motorola downloaded Watson as a result of a recommendation from a colleague. This is particularly encouraging given our initial communication efforts focused on a fairly narrow group within Motorola, and that referrals are typically based on favorable impressions. One user said, "After checking with [Management] on how broadly we could share the tool, I decided to send instructions for downloading the software to all the sales and marketing, operations and development associates in our group." We sent a survey to users who registered for the Motorola version of Watson in order to find more information about how they were using the system. Of the 92 users registered, 16 replied to the survey.

Those who responded mostly reported success using Watson. In the survey, we asked how often Watson was used. Most users reported they made use of Watson 1-5 times a week, though some used it more frequently. When asked why their usage was at this level, instead of higher, users said they were spending more time reading information retrieved from their research efforts instead of searching for sources. One Business Intelligence researcher said, "If I gather content on a subject, I still have

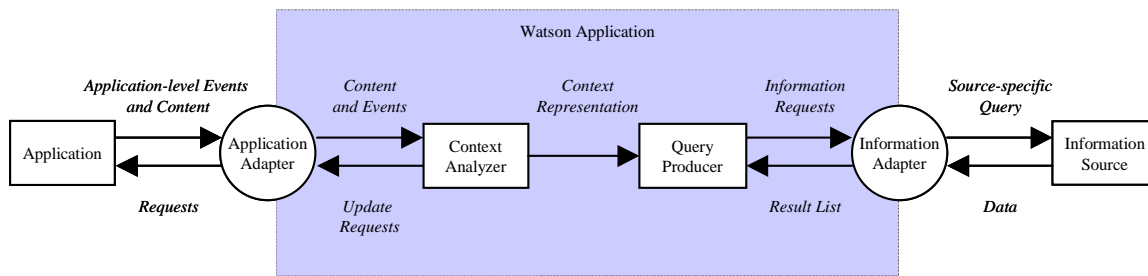


Figure 3: Watson Architecture

to spend some time with the material in order to use it in presentations or papers we are creating for our internal customers.” Thus more frequent use of Watson may not be possible given the time constraints on the tasks in which Watson is most frequently used.

When asked directly whether the information Watson retrieves is useful to them, two-thirds of the respondents rated the utility of the results returned as 4 or 5 on a 5-point scale. Several users reported they found results they think they would not have found using standard tools. One user said, “The web tab identified some news sources I was not familiar with ... they are helping me a lot.”

Though the survey results represent a small portion of the active Watson users at Motorola, the relatively large number of referrals generated by the release of Watson to the business intelligence community (and its continued use over the long-term) speaks to its success in a real-world environment.

Opportunities for improvement

Face-to-face interviews with business intelligence researchers, and email exchanges with the same, uncovered several opportunities for further development by extending the system with additional features.

Users requested the following improvements:

- (1) Watson should allow users to customize which information resources it searches, including local disks, shared drives, and email. In addition, Watson should allow users to sort information retrieved by date.
- (3) Business intelligence research occurs over a time period that makes it appropriate for Watson to automatically monitor sources for new information as it arises. Users should be able to save a context as an alert, causing Watson to repeatedly query information sources and preemptively notify users of new information as it becomes available.
- (4) Watson should automatically highlight the most relevant sections of the documents retrieved. Research reports are sometimes lengthy and only contain one or two sections of highly relevant material. BI researchers often print out, manually skim, and highlight the most relevant sections in sets of retrieved documents.
- (5) Watson should automatically recognize names of businesses mentioned in text. BI research frequently centers on specific companies. Lists of recognized

company names could be presented in a drop-down list associated with the search in context text entry box, allowing users to easily execute contextually-grounded searches on company names recognized. In addition, certain information sources used by BI researchers, such as EDGAR online⁴ an index of SEC filings, and Hoovers company profiles⁵, are only indexed by name or company ticker symbol, and hence require company names as input in order to be used.

AUTOMATIC CONTEXT TRANSFORMATION AND QUERY FORMULATION STRATEGIES

The above deployment experiences motivated several improvements to the core text analysis and query formation component in Watson. They show that given the constraints of real-world tasks, the system can produce significantly more useful information by going beyond simply retrieving documents that are similar to the one the user is manipulating. This section collects and generalizes the methods we have developed for transforming context representations in order to retrieve more useful information. All of these strategies (except the last two) were implemented as part of the pilot development process described above.

Context analysis and query transformation strategies

Recognize and retrieve: The system can have a list of patterns that, when matched, cause specific information to be retrieved from specialized sources like product catalogs and relational databases. These patterns may include lexical variations of database entries or be simple detectors. This functionality was specifically useful in the Ford deployment, which involved integration with many structured databases.

Query augmentation: Queries can be augmented with terms that restrict results to specific relevant classes of documents, e.g., budgets, business plans, lesson plans, or syllabi. Such functionality ensures relevant types of documents are retrieved and presented appropriately. This functionality was specifically useful in the Bighcalk pilot, in order to satisfy role-specific information needs, e.g., the retrieval of lesson plans for teachers.

⁴ Available at <http://www.edgar-online.com/>

⁵ Available at <http://www.hoovers.com/>

Translation: Translation tables can be used to transform queries using simple pattern substitution rules so they produce more relevant results. For example, internal product names included in queries can be “externalized” before sending them to external sources. This was specifically useful in the Ford implementation, in which documents frequently contained terms unknown to external sources (e.g., Internet search engines).

Source-specific transformations: Stop lists are usually based on statistical properties of the corpus over which a search is executed. The most frequent words are eliminated because they have no discriminating value in a given collection of documents. It is obvious different collections will have different frequency distributions, and, hence, Watson’s adapters should have different stop lists tailored to the sources they encapsulate. Of course, stop lists are only one such useful transformation; others may include translation or augmentation (as described above). This was specifically useful in the Motorola deployment, in which internal knowledge management databases rejected queries that contained common terms used within Motorola (e.g., “Motorola,” “wireless,” etc.).

Role-based activation of sources, analysis strategies and query transformations: The system should configure its context analysis and query formation routines based on the user’s role within an organization. The system can tailor its queries to general-purpose information sources (e.g., Google) in order to retrieve specific classes of documents (e.g., lesson plans) that are of known relevance to the user. Likewise, the sources it queries should also be tied to the user’s role. This functionality (combined with the query augmentation strategy described above) was specifically useful in the Bigchalk deployment to support teachers and students uniquely. In addition, it will be implemented within future enterprise deployment in larger-scale organizations that have specialized roles that require specific kinds of information.

Activation of sources, analysis strategies, and query transformations based on content: The system should further refine its context analysis and query formation routines based on the content of the user’s document. For example, if the document is a technical medical document, the system should search medical journals. This functionality was not implemented in any of the above deployments, but has been previously tested and found effective in the laboratory [8].

Activation of sources, analysis strategies, and query transformations based on task state: The system should configure its context analysis and query formation routines based on the user’s task state (e.g., current step in a multi-step process). This can be facilitated by the development of lightweight tools that allow the user to associate a task with a document, and then indicate when they progress from one step to another. While not implemented, this strategy was directly motivated by the problems associated

with K-12 students using Watson as an information tool, as described above in the Bigchalk deployment.

System Architecture and Implementation

The Watson architecture was revised given the need to implement the above strategies. The core organization of the Watson system continues to follow an adapter design pattern [5]; that is, various information sources and applications communicate with Watson through adapters.

Figure 3 presents the basic components in the revised architecture. As before, application adapters interface with existing applications to extract the content of the user’s current document. Watson’s context analyzers then perform a statistical analysis of the document the user is currently manipulating in the context of how it is being presented in the user’s application. Its query producers formulate queries to information retrieval systems that retrieve related documents. These queries are sent to a variety of information sources, each accessed via an information adapter. The results are returned, analyzed further, and presented to the user.

Application Adapters

Application adapters are responsible for determining contextual properties of the document text including (1) the user’s focus of attention, as determined by the active selection, insertion point, or the visible portion of the document and (2) the form of a given span of text, in particular, the line height of a span as compared with the modal line height of text as it appears in the document. Application adapters eliminate background text, including (1) structural elements, such as text occurring in menus and other navigational elements (in Web pages) and (2) text that occurs in headers, footers or signatures. This heuristic analysis of the user’s context produces typed spans of text which are then transmitted to the Watson engine.

The Watson Engine

Internal components of the system, such as content analysis and query generation modules, are coordinated using a blackboard architecture [6, 9]. Information processing components coordinate activity through a hierarchical shared memory, or blackboard. The shared memory is divided into a set of nodes. Each node contains a property list (a table of name-value pairs) that store the information contained within that node. The nodes also contain a pointer to their parent, and a list of listeners. When a property in a node is modified, added, or removed, the listeners at that node and the listeners at all of its parent nodes receive a notification event.

Each active application context is represented by a session node. Session nodes correspond with an active application window, e.g., a document being edited in Microsoft Word. Sessions group sets of documents or document versions viewed within that Window (e.g., revisions to a single Word document). Document objects contain the content of

the document being edited or viewed as a list of spans of text in a given style.

Each document has an initial recommendation context, which corresponds to the set of recommendations first presented to the user. Documents can have multiple contexts, generated, e.g., when the user performs a search in context. Documents contain derivative representations of themselves, e.g., a word histogram. They also group sets of queries generated by query producers, which operate on the aforementioned representations. Queries represent each information request, formulated in an abstract query language, specifying a given information goal. For example, one information goal could be to retrieve a company dossier. Queries group sets of results returned by executing the query on appropriate information sources.

Each information-processing component has its own thread of control, which processes messages received through the hierarchical shared memory as a result of property change events within the nodes. Information processing components each serve a single information processing function. They implement a simple, uniform interface, and therefore can be loaded at run time. This plug-in model allows the system's functionality to be easily adapted to user requirements after deployment. The organization and behavior of information processing components determines the behavior of the system as a whole. Thus, to extend or modify the behavior of the system, one need only implement and load another information processing component.

The first major class of information processing components is context analyzers, which produce representations of the user's context.

The second major class of information processing components is query producers. Query producers are responsible for transforming representations of the user's context into a set of information goals or queries.

Context analyzers

The system's representation of the user's active context is determined by context analyzers. The content of the user's document is sent via application adapters to the Watson engine, which is further analyzed by the following information processing components.

The Histogram Context Analyzer computes a histogram of terms occurring in the content along with their presentation attributes as computed by the application adapter. Thus one form of context representation is a list of terms, their frequencies, and their presentation styles (normal, emphasized, and de-emphasized).

Another form of context representation includes a list of detected entities in the text of the user's document. Using detectors compiled from existing databases, the Entity Context Analyzer asserts the presence of a given entity.

Two Entity Context Analyzers have been implemented: one detects street addresses, the other, names of publicly-traded companies.

In addition, a classification of the active text in a set of known document classes could form yet another context representation.

Query producers

Query producers operate on the above context representations in order to form information requests.

The Similar Documents Query Producer uses the statistical representation from the Histogram Context Analyzer to select terms to include in a query. Each term is weighted according to its frequency and presentation style, and given zero weight if it occurs on a stop list (as in [2]). Terms with weights greater than or equal to the median weight are selected for inclusion in the query. The resulting Similar Documents Query is then sent to vector-space retrieval engines.

The Translating Similar Documents Query Producer uses the same process, but applies substitutions defined by a predetermined translation table. This translated query is then sent to external resources, in deployments for which this is appropriate.

The Augmented Similar Documents Query Producer uses a predetermined augmentation strategy (e.g., prepending terms aimed at retrieving a specific type of document) to augment the similar documents query, as above, in order to retrieve documents of a specific type.

Detected entities (and their types) are used by Entity Query Producers, in order to formulate queries that retrieve records from a predetermined list of data sources. Each Entity Query Producer relates 1-1 with the type of entity and the data source to be queried. For example, the Company Info Query Producer generates queries to information sources that provide company information for each company name detected.

The Phrases Query Producer uses the output of the Histogram Context Analyzer to determine the salient phrases that occur in the user's document, based on a similar weighting scheme, in which the weights of terms in a frequently-occurring sequence are used to determine the phrases' overall weight. The resulting phrases are then sent to sparsely-indexed information sources, such as image and product catalogs.

Information Source Adapters

The information source adapter architecture has been updated to reduce the complexity of wrapper development by providing an XML-based wrapper implementation language with task-specific primitives. Each information source adapter describes what the information source requires as input, and what it produces as output, in terms of named types that correspond to the query types produced by the above analysis. Based on this information, the

system automatically executes a given query on only the appropriate data sources.

Benefits of the architecture

First, the uniform application of the plug-in model throughout the architecture allows the functionality of the Watson system to adapt to the diversity of users' needs (as presented earlier). In addition, as organizations change and adapt business process, the application can reflect these changes with minimal development effort.

Second, as external information systems and user applications evolve and change, Watson can adapt to the changes without requiring a full-scale redeployment. Likewise, application adapters can be modified or created as needed.

CONCLUSIONS

The work presented here demonstrates the utility of software deployment as a means of moving research forward by introducing new research problems grounded in real-world experience. Northwestern's Center for Technology Commercialization has been created to facilitate the development of information technologies (such as Watson) beyond that which is possible using traditional means, providing valuable opportunities for researchers to understand and address new and relevant research problems.

As part of this process, we were able to observe the Watson system working in several real-world settings. This led to the discovery of a number of interesting and relevant research problems, and the development of the techniques presented above.

ACKNOWLEDGEMENTS

The authors would like to thank David Leake for helpful comments on this paper, Rich Gilbert, Kim Imbrogno, Vicki Leonardo, Joe Goldberg, Mark Hatfield, Robbie Higgins, Tom Krause, and others who helped with the pilot deployments at client sites, and Michael Ruberry and Harrison Stein, who contributed to the Bigchalk pilot.

REFERENCES

1. Budzik, J., Hammond, K. J., Birnbaum, L., and Krema, M., "Beyond Similarity," in Proceedings of The AAAI 2000 Workshop on Artificial Intelligence for Web Search, (Austin, TX, USA), AAAI Press, 2000.
2. Budzik, J., and Hammond, K. J., "User Interactions with Everyday Applications as Context for Just-in-time Information Access," in Proceedings of The 2000 International Conference on Intelligent User Interfaces, (New Orleans, Louisiana, USA), ACM Press, 2000.
3. Budzik, J., Hammond, K., and Birnbaum, L., "Information Access in Context," Knowledge Based Systems, 14(1-2), 37-53, 2001.
4. Budzik, J., Bradshaw, S., Fu, X., and Hammond, K. J., "Supporting on-line resource discovery in the context of

ongoing tasks with proactive software assistants," International Journal of Human-Computer Studies, 56,47-74, 2002.

5. Gamma, E., Helm, R., Johnson, R., Vlissides, J., and Booch, G., "Chapter 4: Structural Patterns," Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley, 1994.
6. Hayes-Roth, B., "A Blackboard Architecture for Control," Artificial Intelligence, 26(3), 251-321, 1985.
7. Kulyukin, V., "Application-Embedded Retrieval from Distributed Free-Text Collections," in Proceedings of The Sixteenth National Conference on Artificial Intelligence, AAAI Press, 1999.
8. Leake, D., Scherle, R., Budzik, J., and Hammond, K. J., "Selecting Task-Relevant Sources for Just-in-Time Retrieval," in Proceedings of The AAAI-99 Workshop on Intelligent Information Systems, AAAI Press, 1999.
9. Maglio, P., Barrett, R., Campbell, C., and Selker, T., "SUITOR: An Attentive Information System," in Proceedings of The 2000 International Conference on Intelligent User Interfaces, (New Orleans, Louisiana, USA), ACM Press, 2000.
10. Pazzani, M., Muramatsu J., and Billsus, D., "Syskill & Webert: Identifying interesting Web sites," in Proceedings of The Fourteenth National Conference on Artificial Intelligence, AAAI Press, 1996.
11. Price, M., Golovchinsky, G., and Schilit, B., "Linking By Inking: Trailblazing in a Paper-like Hypertext," in Proceedings of HyperText 98, (Pittsburgh PA USA), ACM Press, 1998.
12. Pyzdek, T., The Six Sigma Handbook. New York, NY, USA: McGraw Hill, 2000.
13. Rhodes, B., and Starner, T., "A continuously running automated information retrieval system," in Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology 1996.
14. Rhodes, B., "Margin Notes: Building a Contextually Aware Associative Memory," in Proceedings of The 2000 International Conference on Intelligent User Interfaces, (New Orleans, Louisiana, USA), ACM Press, 2000.
15. Salton, G., and Buckley, C., "Improving Retrieval Performance by Relevance Feedback," in: Spark Jones, K., and Willett, P., ed., Readings in Information Retrieval. Can Francisco, CA: Morgan Kauffman, 1997.
16. Spink, A., Wolfram, D., Jansen, B. J., and Saracevic, T., "Searching the web: The public and their queries," Journal of the American Society for Information Science, 53(2), 266-234, 2001