An Integration Architecture for User-Centric Document Creation, Retrieval, and Analysis

René Witte

University of Karlsruhe, Faculty of Informatics,
Institute for Program Structures and Data Organization (IPD),
Karlsruhe, Germany
Email me@rene-witte.net

Abstract

The different stages in the life-cycle of content—creation, storage, retrieval, and analysis—are usually regarded as distinct and isolated steps. In this paper we examine the synergies resulting from their integration within a single architecture.

Our goal is to employ such an architecture to improve user support for knowledge-intensive tasks. We present a case study from the area of building architecture, which is currently ongoing.

1 Introduction

Dealing with the overwhelming amount of information readily available today is one of the biggest challenges in computer science. A number of different research areas already deal with knowledge management during the distinct stages of its life-cycle: content creation (document management systems), storage and retrieval (digital libraries, information retrieval systems), and analysis (information extraction systems, text mining).

In this paper we claim that this distinction should be abandoned in favour of an open architecture that incorporates a continuous and iterative life-cycle of content creation, retrieval, and analysis. Before we analyse the requirements for such an architecture in more detail, we first introduce a case study from the application area of building architecture, which is currently in progress. We then present the first version of a system architecture supporting the detected requirements.

Case Study: Analyzing a Historical Encyclopedia of Architecture

Our ideas are perhaps best illustrated within the context of a project we just started: the analysis of a comprehensive multi-volume encyclopedia of architecture written in German in the late 19th and early 20th century.[1] Two user groups are involved in the analysis within this project: *architects* and *building historians*.

For contemporary architects, the encyclopedia represents a large, untapped knowledge source, especially for the restoration and reconstruction of buildings from the same time period. Building historians, on the other hand, are interested in a detailed analysis of an author's work and the context of its creation. The encyclopedia, for example, contains many descriptions of idealized principles and processes, whereas actual constructions often followed rather different, pragmatic ideas. Thus, without a detailed prior analysis by a domain expert (in this case building historians), the knowledge stored in the encyclopedia cannot be safely applied. And even information deemed "correct" may lead to wrong results when applied without a detailed analysis of the author's specific context: In our example on architecture, certain materials may not be available anymore or exhibit vastly different qualities, tools and processes that were common hundred years ago—and thus not explicitly described—are often lost. All this makes it impossible for the contemporary user to naïvely apply the stored knowledge.

Thus, system support cannot stop after information has been delivered to the user. Ideally, all the tasks outlined above are available through the user's web browser, as naturally as viewing yet another page.

2 Requirements

Based on the case study we identified two major requirements systems must fulfill to better support

Copyright ©2004 René Witte. Some rights reserved. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/2.0/de/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

[1] Edited by Joseph Durm ⋆4.2.1837 Karlsruhe, Germany, †3.4.1919 ibidem.

users in dealing with complex, large-scale information sources.

2.1 Integration of Content Creation and Retrieval

As illustrated above, the common static view of a "finished" document that is to be retrieved, viewed, and used by a reader is not sufficient to adequately support knowledge-intensive tasks: Users must be able to add their own information to a knowledge source; in our example, a building historian might want to add a detailed analysis to a chapter of the encyclopedia. Another user, maybe an architect, might want to annotate a section with experiences gathered from the restoration of a concrete building.

While practically all documents are available on or through the Web, its hypertext capabilities are currently not used to directly modify and annotate existing information (books, papers, web pages, etc.). Rather, once content is deemed "completed" it is stored in some kind of archive (e.g., a digital library), from which it is eventually retrieved as a monolithic entity, used for the production of yet more content.

Moreover, the task of information retrieval [1, 8] is typically not integrated with the task of content development. Rather, the user has to retrieve documents he believes are required for his task and then base content development on the information found. While a new document search can always be initiated manually, it is a much more compelling view that content development and retrieval could be integrated: a system that continually scans and analyses new text entered by a user should be able to search additional relevant information and present them to the user, who could then inspect the new data, integrate it, add cross-references, or reject the proposed sources.

Another important point of the case study is that knowledge from a source cannot be applied without a description of the context of *both* a document's creator and its reader. Only an explicit representation of the two context frames allows for a (semi-)automatic translation between them; in our example, we have to adapt over 100 year old knowledge to modern standards and vocabulary, but similar problems will increasingly appear in the medium and long-term future, when all the documents that are currently created and stored in digital form become "historic knowledge" themselves.

2.2 Integration of Information Retrieval and Natural Language Analysis

Currently, users obtain documents through some kind of indexing and ranking systems: web search engines for plain web pages, or some kind of information retrieval systems for digital libraries (historically, these system come from different roots, but modern implementations exhibit some kind of overlap between these techniques). In either case, the systems always return *complete* documents, be it web pages, papers, or whole books. This is one of the primary reasons behind the feeling of "information overload" shared by so many users: with a virtually endless source of information and seemingly relevant documents, how can one be sure to have not only found, but also read and absorbed all the important information? In our case study, just the encyclopedia by itself already comprises more than 40 volumes, which is far too much information to scan without automated assistance.

Our approach is to supplement basic information retrieval with tools from the area of natural language processing (NLP), like text mining and information extraction. While a complete understanding of natural language texts is not even remotely a possibility, there now exists a number of robust NLP tools that can contribute to a semantically richer understanding of a large set of documents: document classification and clustering, automatic summarization, named entity recognition and tracking, and co-reference resolution. Although each of these approaches has a number of deficiencies and limitations, they nevertheless can provide information that are much quicker to scan and absorb than the complete source; an example is the keyword-style summarization of a (newspaper) article in just 10 words [14].

Thus, our core idea here is to combine the standard document retrieval and presentation systems with a natural language processing component that contains a number of specialized analysis tools.

3 Architecture

We now present the architecture we developed to support the detected requirements, as it is currently being implemented. It is based on the standard multi-tier information system design. Its primary goal is to integrate document retrieval, (semi-)automated analysis, and content annotation as outlined above.

Figure 1 shows our integration architecture with its main components. We now discuss each of the four tiers in detail.

Tier 1: Clients

The first tier provides access to the system, typically for humans, but potentially also for other automated clients. In the first version, users will access the system via a standard web client. Additional "fat" clients can be added as well, for example a word processor.

Tier 2: Presentation and Interaction

Tier 2 is responsible for information presentation and user interaction. In our architecture it has to deal with both content development and visualization.

A model that proved to work surprisingly well for cooperative, decentralized content creation and editing is the idea of a *Wiki* (or *WikiWikiWeb*), where every

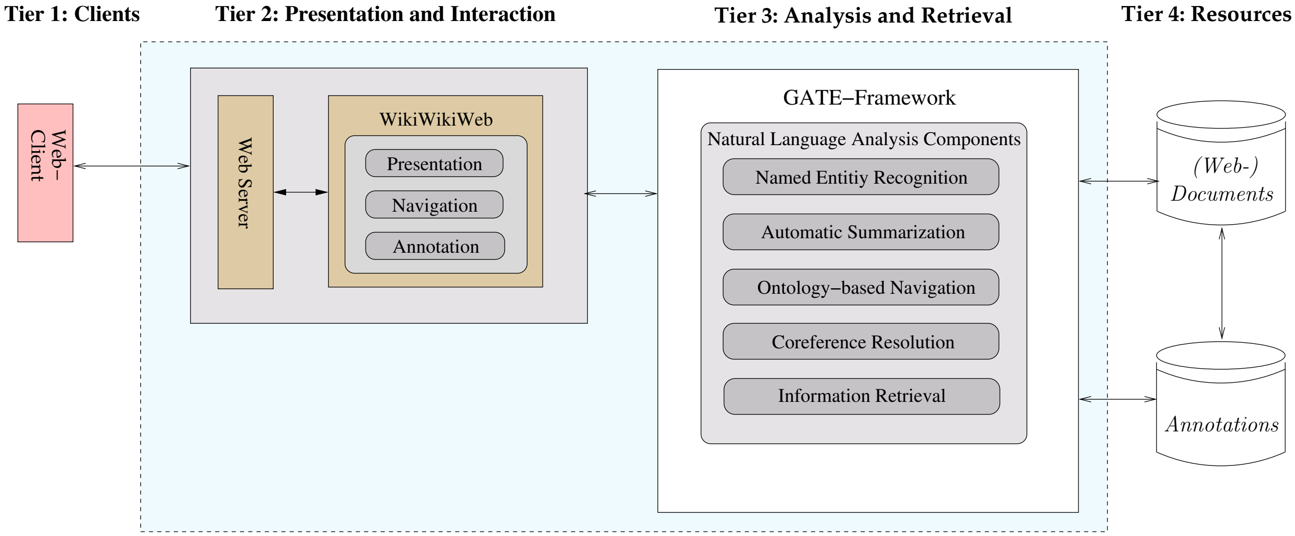

Figure 1: Integrating content development, retrieval, and analysis within a single architecture

user is allowed to create new and edit existing information [10]. Traditionally, Wikis have been used to develop new material (most prominent example being the online encyclopedia *Wikipedia*), but our approach here is to combine both existing (e.g., books, papers) and user-provided content within the same architecture by integrating (and enhancing) one of the freely available Wiki engines.

Tier 3: Retrieval and Analysis

Tier 3 provides all the document analysis and retrieval functionalities discussed above.

The natural language analysis part will be based on the GATE *(General Architecture for Text Engineering)* framework [4, 5], one of the most widely used NLP tools. Since it has been designed as a component-based architecture, individual analysis components (called *processing resources*) can be easily added, modified, or removed from the system. Some of the analysis components we plan to include are:

Classification and clustering of documents, based on the *Bow* toolkit [11];

Ontology-based navigation and retrieval using WordNet [6] for open domain content and specialized ontologies for specific domains [7];

Co-reference resolution for identifying entities within and across documents that refer to the same object under different names [13, 14];

Automatic Summarization to provide keyword-style or full-paragraph summarizations of either single documents or document clusters, which we developed in the ERSS system suite [2, 3].

These higher-level components in turn rely on many other low-level natural language processing resources, including tokenization, named entity recognition, part-of-speech (POS) tagging [9], noun phrase (NP) chunking, predicate-argument extraction [12], among others. Some of these come as basic building blocks with the GATE system, others are readily available as open-source modules. Additional domain-specific components can also easily be added, for example a context management and translation module for our historical architecture.

A final consideration is how to integrate the results of these automatic analyses with the original content and the user-provided annotations. Since we aim to provide an integrated view of all a document's facets, we decided to add them to a document's annotations as well, i.e., they are treated similarly to the information added by a human through the Wiki component.

Thus, original content, human and machine annotations constitute a combined view of the available knowledge, which forms the basis for the cyclic, iterative create-retrieve-analyze process.

Tier 4: Resources

Resources (documents) either come directly from the Web (or some other networked source, like emails), or a full-text database. For our case study, the encyclopedia will be stored in a database holding the scanned page images as well as the OCR'd information. The GATE frameworks provides the necessary resource handlers for accessing texts transparently across different protocols.

4 Conclusions

Making the Web more "intelligent," supporting users by providing services that are semantically richer than plain document indexing and presentation, is a very active and diverse research field [15].

Though the idea of a text mining system, which is an important part of our architecture, is not new, they have so far primarily been deployed only within proprietary business settings or confidential government-funded intelligence services. We predict that such systems will become more widespread in the future and part of our day-to-day web experience, and also be integrated with other (office) tools.

Our main contribution therefore lies in the integration of several techniques for knowledge management that have traditionally been seen as separate, distinct activities: content creation and editing, storage and retrieval, as well as content mining and analysis. We believe that the integration of these activities within a single architecture and within a cohesive, iterative process will provide a user with more support than it is possible with the current separation into isolated tasks.

Another important aspect of this work is the evaluation of NLP-enhanced tools within a real-world scenario. While the NLP community already has moved towards standardized task evaluations (see the NIST-sponsored TREC, MUC, and DUC competitions[2]), there is still no agreement on precisely how much such tools can contribute within a larger, real-life, knowledge-intensive project. We hope to achieve an insight on this issue through our collaboration with building historians and architects.

Finally, we also plan to evaluate the architecture within two other projects: *software reverse engineering*, where it will provide a unified view of a system's software documentation, source code analysis, and requirements specification, and *bioinformatics*, where it will analyze web pages and research papers on fungal genomics.

Acknowledgments. The *Durm* encyclopedia project is funded by the DFG. Stefan Hänßgen and Lars Wetzel provided helpful suggestions for this paper. Natural language processing capabilities are delivered in part by Sabine Bergler's CLaC group *(Computational Linguistics at Concordia)* in Montréal, Canada.

References

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley, 1999.

[2] Sabine Bergler, René Witte, Michelle Khalife, Zhuoyan Li, and Frank Rudzicz. Using Knowledge-poor Coreference Resolution for Text Summarization. In *Workshop on Text Summarization*, Document Understanding Conference (DUC), Edmonton, Canada, May 31–June 1 2003. NIST. `http://duc.nist.gov/`.

[3] Sabine Bergler, René Witte, Zhuoyan Li, Michelle Khalifé, Yunyu Chen, Monia Doandes, and Alina Andreevskaia. Multi-ERSS and ERSS 2004. In *Workshop on Text Summarization*, Document Understanding Conference (DUC), Boston Park Plaza Hotel and Towers, Boston, USA, May 6–7 2004. NIST. `http://duc.nist.gov/`.

[4] H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002. `http://gate.ac.uk`.

[5] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[6] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[7] Dieter Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce.* Springer-Verlag, 2nd edition, 2004.

[8] Reginald Ferber. *Information Retrieval.* dpunkt.verlag, 2003.

[9] Mark Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong, October 2000.

[10] Bo Leuf and Ward Cunningham. *The Wiki Way, Quick Collaboration on the Web.* Addison-Wesley, 2001.

[11] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`, 1996.

[12] Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. Using predicate-argument structures for information extraction. In *41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 8–15, Sapporo, Japan, July 7-12 2003.

[13] René Witte. *Architektur von Fuzzy-Informationssystemen.* BoD, 2002. ISBN 3-8311-4149-5.

[14] René Witte and Sabine Bergler. Fuzzy Coreference Resolution for Summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24 2003. Università Ca' Foscari. `http://rene-witte.net`.

[15] Ning Zhong, Jiming Liu, and Yiyu Yao, editors. *Web Intelligence.* Springer-Verlag, 2003.

[2] These focus on the tasks of information retrieval (TREC), message understanding (MUC), and document summarization (DUC).