A new structure for news editing

by D. Gruhl
W. Bender

Ideally a computational approach could assist in the human-intensive tasks associated with selecting and presenting timely, relevant information, i.e., news editing. At present this goal is difficult to achieve because of the paucity of effective machine-understanding systems for news. A structure for news that affords a fluid interchange between human and machine-derived expertise is a step toward improving both the efficiency and utility of on-line news. This paper examines a system that employs richer representations of texts within a corpus of news. These representations are composed by a collection of experts who examine news articles in the database, looking at both the text itself and the annotations placed by other experts. These experts employ a variety of methods ranging from statistical examination to natural-language parsing to query expansion through specific-purpose knowledge bases. The system provides a structure for the sharing of knowledge with human editors and the development of a class of applications that make use of article augmentation.

A news editor fine-tunes and prioritizes information based on criteria that include timeliness, importance, and relevance to the audience.

—Jack Driscoll

News editing is an exacting problem. Many factors contribute to making the selection and presentation of timely, relevant information a task as daunting as it is necessary. People want to be kept informed of events and occurrences that affect them, but at the same time they do not want to wade through the tens of thousands of news articles available every day to find what they need.

And it is not just a matter of deciding which articles may be of interest. How much is enough and how much is too much is a delicate balance to strike. One Elián Gonzales article a week might be interesting—fifty might not be. A person's source of news needs to express what is new rather than just what has happened. However, if someone has relatives in Pakistan, then *every* article about a revolution that occurs there may be of interest.

The amount of time that the average person can spend on the news each day is more or less fixed. Consequently, many personal and situational interests compete for this time. Decisions must be made as to what to present, in what order, and in what way. It requires understanding on the editor's part of not only what a given article is about, but also what the context is, or how the particular article relates to other articles that are available, as well as how it relates to the reader.

The task of an editor, then, is to examine the news for a given day and try to find the *meaning* in it—that is, not only to understand the article, but also to understand its context. What is new or timely? What is of importance? What is of high general interest? What does the reader need to know about? What would the reader like to know about? What informs, educates, guides, or entertains? How many articles on a topic are appropriate, and, if the answer is not "all of them," then what should be kept and what discarded?

Advantages of a human editor

When considering an on-line newspaper as a primary news source, it makes sense to consider what editors do, what issues they face, and what their strengths are, as well as their weaknesses.

"In today's *Journal of the American Medical Association* . . . " There is a need to integrate multiple sources of news, since not all articles of interest come from the same source. In fact, unless someone's interests are exactly aligned with the focus of a particular publication, the reader probably will need to consult several sources of information each day to find what he or she needs. For example, the televised weather report in the morning, the newspaper on the bus ride to work, the radio for the outcome of the afternoon baseball game, and an on-line news service for up-to-date stock information: each of these sources presents information in a different format and, if a unified presentation is to be made, all need to be understood and considered together.

"Dear Editor, . . . " Editors do not work in isolation. They receive feedback from the community they serve in a number of ways: direct letters, telephone calls, or electronic mail to the editor; comments from their colleagues; focus and market surveys; and simple hard numbers like newsstand sales when a particular headline is run. This feedback allows the editor to better serve the needs of the community. Note that none of this feedback is actively solicited from the readership. Rather, these are observations that are made passively or as a result of user-initiated comments. There is something to be said for assuming that, if there are no complaints, then something is going right and need not be modified.

On your doorstep. One other aspect of the "real world" editorial process is that there is no waiting. When one reaches for the newspaper there is no delay. The fact that a newspaper may represent a 24-hour production cycle, thousands of person-hours of preparation, and a variety of news sources is inconsequential. When you want the news, there it is. This is especially important when contemplating on-line editorial approaches that require significant processing time. It may seem obvious, but the right time to think about the news is not the first time someone asks for a copy of the newspaper.

The On-Line Times? When we type a query into a search engine, we are making a request that such an engine consider a large number of possible articles and select and present some of those articles for our consideration. This is nothing more nor less than an editorial process. Many search engines return results that would be considered poorly edited. Sometimes they return nothing, providing no explanation of how the request was too restrictive. Other times they return far too many results, swamping the user with a plethora of information to wade through and decide on. Neither of these alternatives is particularly attractive to an end user. Little wonder then that most individuals would be unwilling to accept an on-line computer-generated newspaper when they have the opportunity to read a traditional one, where the selection is done by human editors.

I do not know what I want! Defining searches is a difficult task. It is even more difficult when trying to define what the search should be about. One of the reasons a reader may subscribe to a newspaper is that he or she trusts the editors will provide needed information. An on-line newspaper should provide users with some reasonable starting point, even if they themselves do not know what that is.

I want what he has! One role a newspaper fills is that of providing a sense of community and shared world view. The conversation that starts with "Did you see the front page of the newspaper?" is absent in a world where each newspaper is custom-made for an individual. The common context provided by a shared information source is important, for without it, individuals lack a common reference point with which to engage in discussion.

News as a product/news as a service. Ideally a computational approach could alleviate the human-intensive tasks associated with news editing. At present this goal is difficult to achieve because of the paucity of effective machine-understanding systems for news. A structure for news that affords a fluid interchange between human and machine-derived expertise is a step toward improving both the efficiency and the utility of on-line news. This structure reflects a reorganizing of on-line news distribution around both a production model and a services model. The on-line news product is a reflection of the traditional paper offering. On-line services include: (1) identifying, contrasting, and relating; (2) analyzing, positioning, and verifying; (3) localizing, augmenting, and remembering; (4) contextualizing, connecting, and associating; (5) expressing, storytelling, and transcoding;[1] (6) learning, interacting, and constructing; and (7) marketing, observing, and transacting. Each of these services contributes to the whole but also has

value when offered as a component service. In a distributed but structured architecture, each of these services can be developed and deployed with relative autonomy.

The ZWrap system. This paper examines an approach by which a computer system, ZWrap,[2] can develop rich structure for a corpus of news. Beyond developing an understanding of each news item, the system attempts to find context for each article, examining how it fits into the larger picture of the news. (In this paper, we consider the understanding of an article to be the result of examining the article and identifying features within it. These features may be as simple as the individual words that appear or as complex as the actors and actions they perform in an article or even the bias with which a particular article was written. "Context" refers to how features relate to each other, especially the way in which they tend to occur in a large number of articles. This includes, for example, both the co-occurrence of features and their associations and implications.) The ZWrap system considers what technologies are needed to keep this context current in the face of a changing external world. It examines ways in which users who are not information retrieval experts can share their understanding with the system and act as a source of common sense for it. An overview of the ZWrap system is shown in Figure 1.

Background. Ideally, the ZWrap system should be easy to assemble, easy to maintain, and easy to improve. It should perform the task of developing and discovering meaning in a reasonably efficient and interesting manner. More specifically, it should assist, simplify, and automate the types of editorial decisions that a human editor must face and resolve, in a way that is amenable for use in an on-line news environment.

The news editing task embodies information retrieval, in that searching for relevant articles within a corpus is part of the problem being addressed. But a news editing system must span multiple corpora and multiple domains; it must accommodate feedback from multiple sources; it must be capable of continuous updating; it must facilitate the dynamic redefinition of relevance while maintaining high precision and recall; it must support automatic query generation; and it must be the host for a shared context across a varied audience.

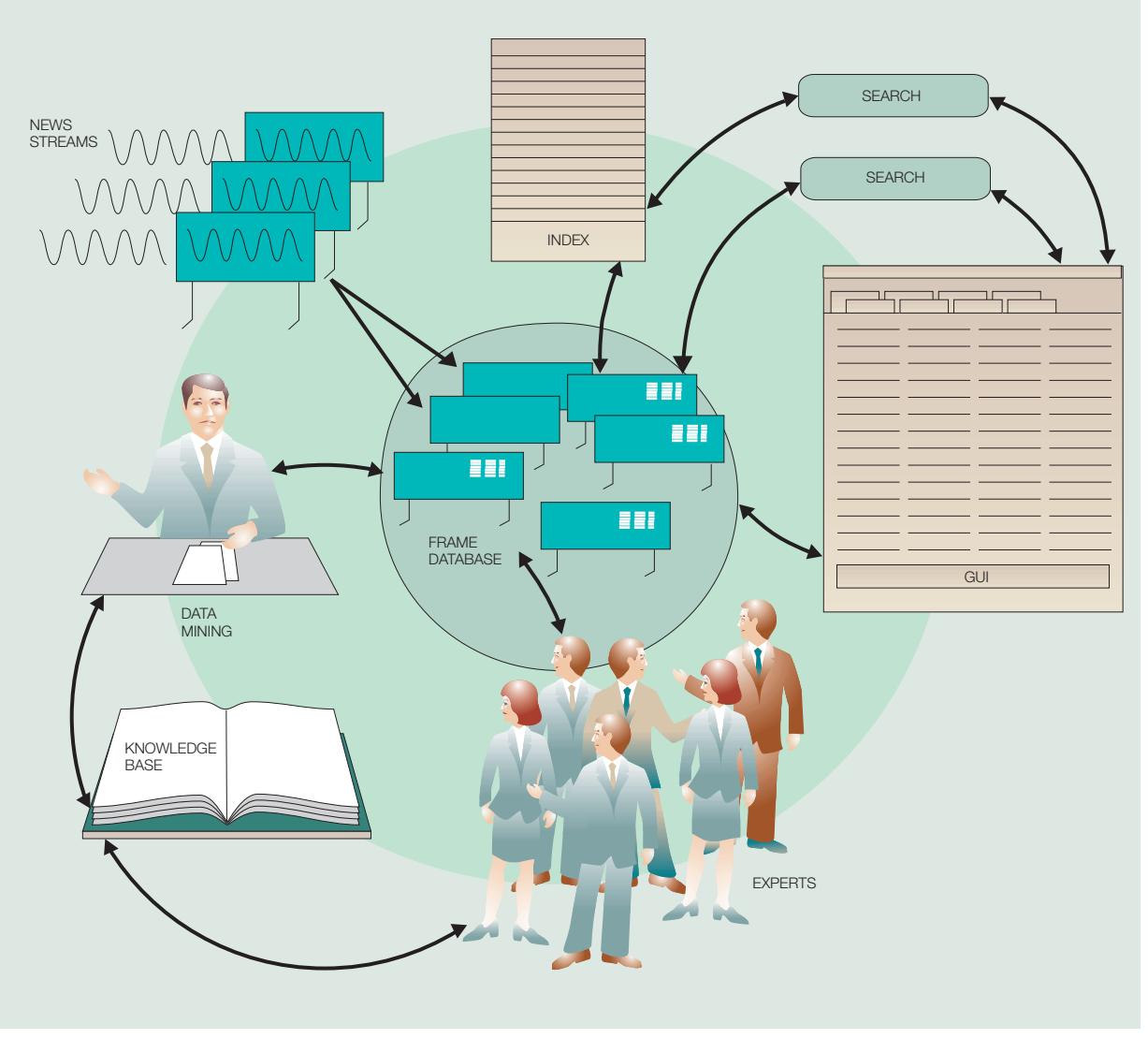
The ZWrap approach to news editing is to provide a structure for interoperation of multiple components rather than attempt to design an algorithm that is optimized along the many constraints of the problem. Components were chosen with little regard to employing the best practice within a particular subfield. Those components (e.g., a classifier or a router) that are found most useful are candidates for optimization.

While the field of information retrieval is well populated, the space of projects similar to the ZWrap system is rather sparse. Examples include Apple Computer's Sherlock**[3] (with its "pluggable" search and selection algorithms), IBM's Lotus Notes** (with its loadable experts and integration framework), the UNIX** shell (that allows many processes to interoperate through pipes), the World Wide Web (if viewed as a single system and when considering CGI [common gateway interface] programs as plug-ins), integrated development environments (such as Metrowerks' Code Warrior**), the Emacs editor (in its guise as an electronic mail reader, spell checker, development environment, document authoring tool, and overall "kitchen sink"), and the Media Bank[4] (with its distributed architecture and community of viewers). All of these approaches seek to perform a higher-level task by marshaling a number of conceptually lower-level operations. The ZWrap system might best be described as an attempt to serve the function of information retrieval within the domain of exploring, selecting, and presenting news.

The Zwrap approach to news editing is to provide a structure for interoperation of multiple components.

Paper organization. In this paper, we address the task of news selection for an on-line environment. Our approach is to create a symbiotic relationship between computer and human editors and human consumers. We accomplish this within the framework of a blackboard structure that manages the simultaneous execution of multiple experts. We conclude with a discussion of the efficiencies gained by this approach, in particular, the advantages of article selection through use of precognition, the use of mod-

Figure 1 Overview of the ZWrap system



ular, domain-specific experts, and an augmented presentation.

In the following sections, these topics are discussed: (1) blackboard systems and the overall architecture, (2) data representation and management, (3) networking, distribution, and parallel computation, (4) searching and sorting in an augmented database, (5) statistical examinations, (6) user interface and presentation, (7) implementation, and (8) evaluation and conclusion.

Blackboard systems

Blackboard systems are an old idea, proposed by Newell in 1962.[5] Roughly speaking, a blackboard system is one that employs a collection of experts that independently look at a blackboard.[6] There is one piece of chalk and when an expert has something to contribute to the understanding of a problem, the expert takes the chalk and writes the contribution on the board. Experts are generally not allowed to talk to each other; all communication is done via the

blackboard. Each of the experts sitting in front of the blackboard is assumed to have a specialized area of knowledge, but may make use of observations written on the blackboard by other experts.

Creating a piece of code to represent an expert is fairly simple. The expert needs to be able to read the blackboard, take the chalk, and write observations on the blackboard. Since all communication is done via the blackboard, no other interexpert protocols are necessary.

Since the experts do not interact with each other except via the blackboard, adding, removing, or changing an expert has minimal impact on the overall system (although, if one expert depends on the work of another, some complications can arise). This allows the development and improvement of experts to be an ongoing process. Since experts do not need to interact except through the blackboard they can all "think" about the problem simultaneously. This opens the possibility of parallel cognition processes.

From a theoretical standpoint, this architecture allows for the development of a "society of agents," as suggested by Minsky,[7] with a number of specialized experts contributing their observations about a problem in the hope of developing some kind of understanding about it. Each of these experts can evolve independently; new ones can be added at any time; and those that are found to be less useful can be dropped.

Despite all these benefits, blackboard systems have fallen into disrepute. Perhaps the biggest difficulty with this architecture is its problems with efficiency. It has been observed that in general, most experts wait on the actions of another expert.[8] If Expert B needs to look at Expert A's observations, then until Expert A makes those observations, Expert B can do nothing but wait. The necessity of a single piece of chalk with atomic locking makes writing to the blackboard somewhat expensive: when more than one expert has something to say there is conflict over the chalk; and when one expert writes with the chalk, the other experts are obligated to reconsider the blackboard in light of whatever is written—see Figure 2A. It has been observed that blackboard systems perform less efficiently than many alternative architectures, e.g., a pipeline.

Why is efficiency such an issue, given that a blackboard system can help to reduce development time, sometimes dramatically? Blackboards have been applied to problems such as speech processing,[9] sonar contact analysis,[10] and fighter aircraft intercept evaluation,[11] all cases where there is a single or small number of problems being considered and there is an element of time pressure (necessitating efficient processing). These applications highlight both the strengths and weaknesses of blackboard systems. (For a more complete discussion of blackboard systems, see Engelmore et al.[6] and Carver and Lesser.[12])

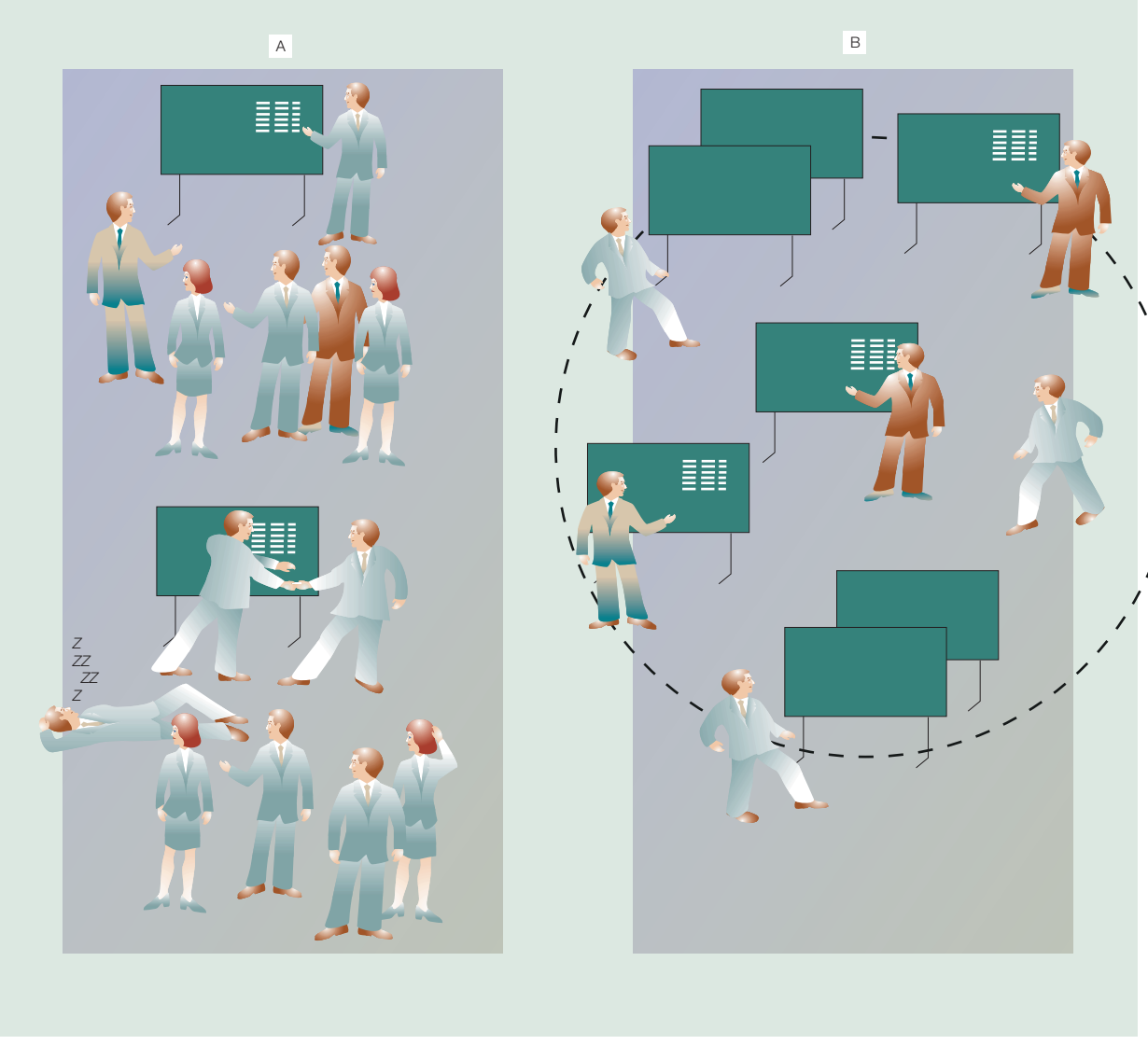
Blackboards and news. Developing understanding of news is a problem that shares many elements with the traditional blackboard problems—it is a complex problem where it makes sense for many experts to work on several different approaches simultaneously. There are, however, a number of key differences that make news an appropriate environment for using blackboards. In the traditional blackboard case, there is a single (or perhaps a small number) of problems being considered, while in news applications, on the order of 10 000 articles arrive daily, thus the number of "problems" is quite large; and unlike the case where a single problem might be relevant for only a matter of minutes or hours, news articles often retain their relevance for days or weeks.

In the context of news, the blackboard architecture can be modified. Instead of several dozen experts standing around a single blackboard, one can instead imagine a room with thousands of blackboards, one for each article. Several dozen experts wander around the room, making observations on each of the problems in process. If a blackboard has an expert standing in front of it, then another expert can just pass it by, coming back to it later when it is free. If each expert has a different color chalk, then those experts that depend on the work of others can just visit blackboards that already have the appropriate colored marks on them. In short, most of the problems with the original blackboard architecture either do not arise or can be avoided in the case of news systems—see Figure 2B.

The blackboard architecture is amenable to the news-as-a-service model described earlier. Experts can be designed to process the news by contrasting and relating article features and by identifying associations between articles.

Implementation. The ZWrap system uses a blackboard architecture for developing its in-frame representation of articles. Experts watch a variety of news sources. Whenever an article arrives, a new

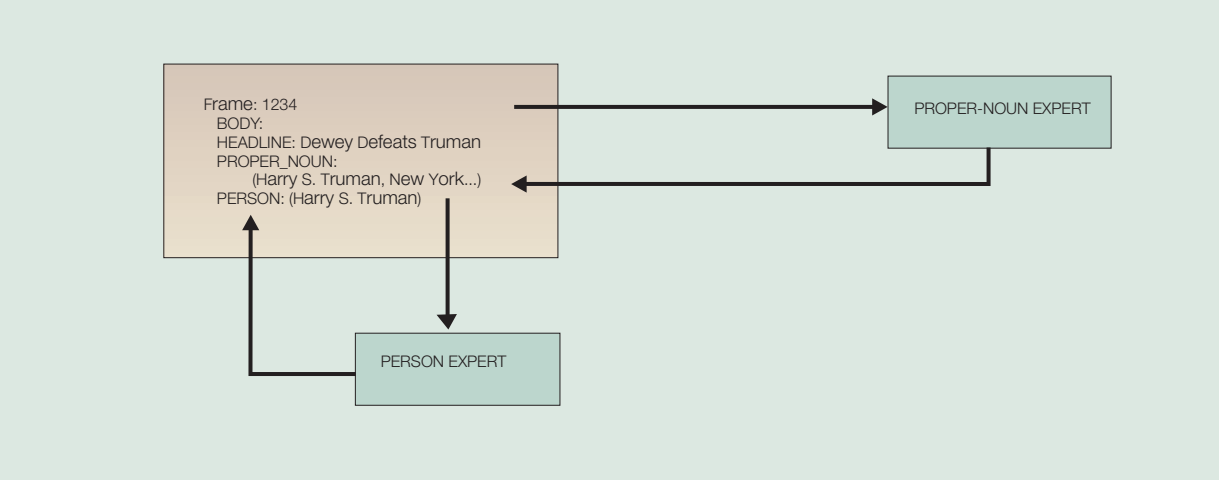
Figure 2 (A) On the top are many experts, all standing around the same blackboard. They all look at the blackboard, and when one has something to contribute, that expert grabs the chalk (if it is available) and writes down thoughts. All communication is done via the blackboard, and only one expert is allowed to write at a time. Below, we see many frustrated experts! (B) With a group of blackboards, experts can wander around, writing on a blackboard that is free.



blackboard is created, the article is posted, and the blackboard is then visited by a number of experts.

In the ZWrap system, one group of experts examines the articles for purely structural elements. These experts extract the headline, the dateline, the author, the body text, the time the article was posted, etc., and place this information on the blackboard, mitigating the problem that different news sources provide news utilizing different structures. These experts ensure that all articles will have more or less the same article elements broken out (e.g., body, headline, date, etc.) and tagged for later processing by other experts.

Figure 3 The proper-noun expert adds a list of the proper nouns found in an article. The person expert examines this list to identify the names of people and writes them into the article.



A second group of experts performs information extraction, parsing, and evaluation on the uniform article elements extracted above. These experts perform tasks such as proper-noun identification, word stemming, date and time spotting, parsing to find references to places, spotting country names, etc. These derived features are written back onto the blackboard representing the article.

A third group of experts uses the derived features to perform high-level augmentation. For example, the person spotter looks at the list of proper nouns and decides (through a simple heuristic) which proper nouns are the names of persons (see Figure 3). The geography expert uses a list of rules applied to the country feature to decide which regions and continents are being discussed in an article. Also included in this group are experts that use these derived features to produce more features. For example, the Media Lab expert looks at the people feature to search for the names of Media Laboratory professors. It also looks for "MIT Media Lab" in the proper noun feature.

Ultimately, this parade of experts takes an article as an initial, monolithic piece of text and transforms it into a richly annotated structure of identified high-level features. These features are used for three different purposes: (1) search engines use them for selecting articles; (2) clustering and data-mining techniques use them for finding patterns and trends in the articles; and (3) display engines use them for presenting articles to the user in context.

About dependency. One design issue with blackboard-type systems is information-dependency management. In Figure 3 the person expert can do no work until the proper-noun expert has visited the article. This leads to inefficiency if the person expert keeps checking blackboards to see if they have been processed.

The ZWrap system supports a simple directed, acyclical graph (DAG) of expert dependency. It does so by maintaining a list (called a dependency scratch space) of which articles have been visited by which nodes in the DAG. A "downstream" node can examine this list to identify which articles have been visited by every expert it depends on.

If a key/value pair that some other key depends on changes, how does the system know to go back and address the resulting inconsistency? In the current implementation of the ZWrap system, inconsistencies are not repaired. This limitation is being addressed in follow-on work.

Data representation

The use of a blackboard system simplifies the question of data representation, since the only well-defined representation needed is that of the blackboard.

This representation needs to support several actions efficiently: the reading of the contents of a particular blackboard, the addition of a note or notes to a given blackboard, and the search of all blackboards for particular notes or types of notes.

The requirements of both flexibility and speed of access argue for the use of frames[13] as the data storage medium. A frame is a collection of key/value pairs, known as terminals, that describe a particular object or concept (in this case, a news article).

Frames have a number of advantages, not the least of which is their flexibility. New key/value pairs can be liberally added and clients looking for a particular datum can ignore those terminals that they do not understand or need. The ability for experts to ignore what they do not understand or need in a frame is important, since it allows the addition of arbitrary experts without the need to modify any other experts in the system.

A feature borrowed from FramerD,[14] a persistent database of frames (framestore) developed at the Media Laboratory, is the use of a universally unique identification number (called an object ID or OID) for each frame. This 64-bit number allows the frame to be referred to in a succinct, unambiguous manner. Since OIDs are never recycled, they can be used as pointers to articles, and it is assured that the articles being pointed to will never change.

In the blackboard context, each blackboard is represented by a frame. Experts examine the frame to see what observations have been made and make their own observations by adding terminal nodes to the frame. A mechanism exists (inherited from FramerD) to allow a blackboard to be locked while an expert is looking at it.

When an article enters the system, a frame is allocated and the article is entered as the only terminal, under the key Text. As experts examine the frame in turn, they add terminals, with successive terminals containing increasingly higher-level information about the article.

Network protocol

Given that the ZWrap system uses a framestore to represent its blackboards, the question arises as to how the experts will communicate with the framestore to read the blackboards and write their annotations. There are characteristics this communication system should have—some implied by the previously made assumptions and others that simply enhance usability:

- Atomic framestore access. Experts must be able to "grab the chalk" while they are writing on the blackboard.
- Efficient framestore access. In many cases, the time taken for framestore access will not be the bottleneck for programs augmenting the articles. Because experts tend to be computationally expensive, extremely fast access is probably unnecessary.
- Lightweight framestore access. The primary concern of someone coding an expert should not be how that expert gets its data. Also, the computation and memory associated with access should be minimal.
- Concurrent framestore access. The blackboard system gains much of its performance through allowing many experts to access the framestore simultaneously. The extension to multiple blackboards suggests that there be a mechanism for multiple experts to work on different blackboards in the store at the same time.

Taken as a group, these characteristics suggest that the central framestore repository be provided as a service. An expert connecting to the service expects the service provider to worry about issues of concurrent access. This lightens the load on experts and those who author them.

Given the prevalence of network-computing environments, it makes sense to consider that such a service might be provided over a network connection. A network approach also allows computationally intensive experts to run on different computers, allowing for load distribution and for more efficient use of available resources.

A side benefit of a network solution is that any program that can obtain a network connection and use the framestore protocol can operate as an expert. Thus, experts can be written in whatever language is appropriate for the processing they perform and run on whatever hardware or operating system is best for their execution.

All that is needed to utilize a networked framestore server is a protocol by which clients can read from, write to, and lock frames. Such a protocol should be:

- Easy to implement. Encoding messages to go over the network and decoding the response should be

simple for the programmer. The easier it is to implement an expert, the more likely it is that many experts will be developed.
- Low computational overhead. The computational load on the expert should be biased toward the task at hand, not toward frame access.
- Low bandwidth. The percentage of bandwidth dedicated to the protocol should be low. Ideally using this protocol will not slow the expert down.
- Extensibility. New commands and structures should be easy to add, without the need to rewrite existing experts.
- Expressiveness. It is key that experts be able to write nearly any kind of observation into the framestore. The protocol should support arbitrary nesting and combination and extensions of the basic data set.
- Human readability. While not strictly necessary, the existence of a text representation is extremely useful. It facilitates debugging and documenting the protocols.
- Robustness. A protocol that works well in light of the peculiarities of packet transport, dropouts, concurrent connections, etc., is important.

A protocol meeting almost all of these criteria existed at the inception of the ZWrap system in the form of Dtypes. Dtypes comprise a simple network protocol for exchanging LISP-like objects. They were designed for situations where there is routine use of data structures with complex interconnections. They include basic elements such as numbers and text arrays, as well as extensions to special-purpose data types. Dtypes were originally developed by Abramson,[15] later extended by Dienes,[16] and subsequently formalized and extended by Haase[14] and Gruhl. The resulting protocol meets all of the above criteria and serves as the underlying communication system for ZWrap.

The Dtypes library has been implemented under IBM AIX* (Advanced Interactive Executive), DEC Ultrix**, Sun Solaris**, Linux**, NeXT** Operating System, Apple Macintosh** Operating System, and Microsoft's Windows NT**, Windows 95**, and Windows 98**. Versions exist in the Java** language, Perl, C, C++, Scheme, and LISP.

The impact of Dtypes on the ZWrap system is subtle but important. It serves as the *lingua franca* for the various components of the system, allowing them to communicate with each other in an unprescribed manner. This results in components of the system being used in ways that were not originally intended. (For example, in the *MyZWrap* application, a weighted search is implemented as a series of calls to an index server.)

There are other protocols that would allow this kind of interaction as well (e.g., XML [Extensible Markup Language]). What is important is not that the communication protocol be Dtypes *per se*, but that it support arbitrary and nonpremeditated interactions.

Expert distribution. In the ZWrap system, experts can run on any machine that supports the Dtypes protocol, allowing them to be implemented in the environment most practical for the tasks they will undertake.

The use of the Dtypes protocol facilitates the types of distributed, concurrent interactions needed to implement blackboard systems. This approach allows the development of experts in an incremental manner without incurring a performance penalty. The ability to run experts on multiple midrange commercial computers allows the use of many computationally expensive experts without the need to resort to expensive special-purpose, supercomputing solutions.

The distribution of experts over a network allows the simultaneous pursuit of very different approaches to representation, increasing the chance that the system will be able to develop interesting and relevant observations for every article.

Searching in augmented framestores

The approach to searching used in the ZWrap system has an advantage over more traditional approaches in that a host of augmentations are available to help direct the search. In this section, the adaptation of traditional approaches to database searching in the ZWrap environment is discussed.

Boolean searches. Boolean searches are a baseline for many search engines (e.g., AltaVista**, LexisNexis**, Gopher, etc.). Searches of this type examine articles to evaluate a Boolean expression describing the presence of certain words. For example, the Boolean expression

("Bill" OR "Hillary") AND "Clinton"

seeks articles that mention the word Bill or the word Hillary and also mention the word Clinton. The ZWrap system implements this type of search using

the operators AND, OR, and NAND. However, the ZWrap system operates on terminals when searching, not words. Since the set of proper nouns is written back into the frame, the above search (in the ZWrap system) is implemented as

```
((PROPER_NOUN."Bill_Clinton") OR
  (PROPER_NOUN."Hillary_Clinton"))
```

But proper noun is not the only feature in the frame of an article. As an example, consider what can be done with the output of three experts. One of the ZWrap experts spots types of food and notes the occurrence under the "food" terminal. A number of experts are inherited from *FishWrap*[17] (an on-line news system that is a precursor to the ZWrap system), one of which spots morbid news stories, e.g., articles where death, serious injury, or grievous harm occur. A geography expert places information on continents mentioned implicitly in an article into the frames. These can be used together as

```
((FOOD."beer") OR (FOOD."wine")) AND
  (FISHWRAP_TOPIC."morbid") AND (CONTINENT.
  "Europe")
```

which finds (mostly) articles about drunken driving in Europe. With a fairly detailed knowledge of what experts are in the system and some clever authoring of search expressions, reasonably complex concepts can be expressed with just Boolean operations on article features.

There are two problems with this approach. First, there is a considerable onus on the user to understand the details of the system, such as what features are being spotted and what their typical values are. Second, there is no concept of how well an article fits a particular topic. It either matches the search expression or it does not. There is very little that can be passed along, for example, to the display engine to help it decide how the articles should be presented.

Weighted searches. A weighted search, such as that used by AltaVista,[18] introduces the concept of "must" (by prepending a "+" to a search term) and "must not" (by prepending a "−"), allowing a simple notion of query weighting. Some features are designated more important than others but once all of the "must" conditions have been met, other terms contribute to the fitness of an article for selection. For example

```
+Bill Hillary+Clinton−Chelsea
```

finds articles that contain the words "Bill" and "Clinton" but do not contain the word "Chelsea." From this set, articles that also mention "Hillary" are considered a better match than those that do not. Search results are sorted by their ranking. Of course, as before, in the ZWrap system the search terms can be any of the derived features.

Activation/evidential searches. The term "activation search" comes from imagining a database of all the terminals with connections to all of the articles that mention a particular feature. A search is performed by activating the stated concepts and selecting those articles that are in turn sufficiently activated through these connections.

The presence or absence of a feature contributes to or detracts from an article's score for selection, as shown in these examples (from explanations generated by ZWrap):

```
The presence of the PROPER_NOUN "Kosovo" strongly
  supports selection.
The presence of the STEMMED_WORD "Albanian" does
  support selection.
The presence of the PROPER_NOUN "NATO" may
  support selection.
The presence of the PROPER_NOUN "United Nations"
  may support selection.
```

To increase readability by the users, a fuzzy mapping to words is used instead of numeric weights, but this is an arbitrary assignment. A very large weighting (i.e., certainty) is used to allow selected features to be "stop" features; selection is prevented if they occur. Activation-style searches are fully weighted searches. The result of this type of search is a list of articles that can be ranked by their level of activation.

Relevance feedback. The ease of construction from examples suggests that relevance feedback[19] might be a useful approach for designing searches: the user performs an initial search to identify articles similar to the ones he or she is seeking; the system looks for similarities between these articles and uses this as search criteria; the user examines the results of this new search and identifies articles that seem most relevant. The user iterates through this process until articles of the sought-after class are found.

Relevance-feedback searches need not be explicit. If the system can observe the user interactions and infer something about which articles were of inter-

est, then this approach can be used to refine news channels without explicit formulation of rankings.

Multiple algorithms. Ng[20] has found that in some contexts a weighted average of the recommendations of several algorithms almost always performs better than a single algorithm for search tasks, i.e., weighing is better than picking. The ZWrap system is well suited to multiple-algorithm approaches.

News channels. The ZWrap system borrows the concept of channels for news presentation from *NewsPeek*,[21] PointCast,[22] and MyExcite.[23] All of the articles in a channel are part of a specific and hopefully well-defined topic. In general, articles are selected for a channel by searching. In the ZWrap system, any search performed by the user is a candidate for redefining as a channel. This allows a user to apply any search skills he or she might have toward automating the editing of his or her newspaper.

Experts. Searching need not be a one-time event. Once a means of finding a particular type of information is developed, it can be turned into a standing request for information (as a channel). From here, there is a clear evolution toward developing an expert. First, a simple query might be developed. Over time, that query might be refined. Commonalities between queries might be formalized into subqueries. If a subquery is sufficiently useful, then it becomes a candidate for being turned into an expert.

Statistics

The ZWrap system makes use of the relationships among frames through a variety of statistical examinations of the corpus, looking for patterns and trends that develop among high-level features.

Developing good searches by hand is effortful. This is compounded by the tendency for topics to "drift" over time. As explained in the previous section, the ZWrap system seeks to capture this work by allowing searches to be turned into news channels, where they can be used for an extended period of time. Statistics can augment search techniques by flagging unusual events, drawing attention to them for further consideration by experts, human editors, or the user.

In order to apply statistical and pattern-recognition techniques to the task of retrieving articles, some mapping is needed between the articles and a vector of features that represent the article. The common mapping is one that takes words that appear in the article and maps them to individual elements in the vector. These vectors are collected into a single matrix, known as a "word document matrix," that represents the corpus (or at least a training set).

When features carry information in addition to words, techniques that work well on word sets work even better on augmented frames. In the ZWrap system, by the time an article is to be examined statistically, additional features have been spotted, computed, or otherwise added to the frame, making it possible to use a "feature document matrix." It is this matrix that is used by experts to provide context for individual articles and to find associations and differences among multiple articles.

Statistics can augment search techniques by flagging unusual events, drawing attention to them for further consideration.

Statistical techniques. Each article in the ZWrap system is represented as a vector of features, where a_i is the feature vector for article i and $a_{i(n)}$ the nth entry of that vector. The mapping of features to entry is arbitrary but fixed for the corpus. For reasons of efficiency, features with insufficient support may be dropped from this mapping (if a feature occurs only once it is not of much help in classification) and, likewise, overly common features may also be dropped (a feature is equally useless if it occurs all the time). For simplicity, the ZWrap system uses Boolean values to represent features. This means that feature vectors are filled with ones and zeros, representing the presence or absence of a feature in a given article.

Clustering. One task that statistical methods perform well is clustering. There are many different clustering algorithms available, ranging from simple K-means to the more complicated simulated annealing. The goal of these algorithms is to take a large number of items and divide them into groups. They often require that the number of groups is fixed initially or modified by a heuristic during the analysis. Some of the algorithms are described here:

- Simple *a priori* occurrence expectation. *A priori* occurrence is a simple but powerful statistical technique. It looks at a domain (e.g., a channel or the entire corpus) and develops *a priori* statistics on feature occurrence. For example, let $\bar{A}$ be the normalized, average article in a channel. $\|A - \bar{A}\|_2$ or $\cos(A \cdot A)$ is then a measure of how "distant" a particular article is from what is typical for the channel. The set of As for all channels represent the typical or *expected* articles for those channels and a new article is compared to these stereotypical articles in order to decide which channel to place it in.

- Related articles. It would be expected that articles covering the same topic would have similar features. Thus, a simple distance metric like cosine angle between the normalized feature vectors would give some sense of how related articles are. This nearest-neighbor analysis allows automatic identification of related articles. It also can be used to find near-duplicate articles, for related articles that are close, but not too close. This is especially true for news streams that tend to repeat stories with small changes from hour to hour. In these cases, just presenting the most recent article is probably sufficient.

- Association-rule data mining. The next step up from simple occurrence is co-occurrence, determining what features occur together frequently in the same article. Association-rule data mining[24] seeks to find the associations between groups of features, for example that $A \wedge B \rightarrow C$, where A, B, and C are particular features in the corpus.

- K-means. An augmented framestore can be used to explain K-means[25] clusters. K-means is run on a set of LSI (latent semantic indexing) dimensionality-reduced vectors generated from the stemmed-word document vectors through singular-value decomposition. The ZWrap system gives the user an indication of why a cluster has been created by revealing those high-level features that are in common among the articles within the cluster.

- Presentation. Clustering is used to decide what articles to present to a user. If several dozen articles are candidates for presentation within a particular topic, one approach is to cluster them and select the representative articles from each cluster for presentation. The user asks the system to select an article to indicate that he or she is interested in its associated cluster.

- Cluster management. Most clustering algorithms operate on a fixed number of clusters. This is a difficult number to determine if there is no *a priori* reason to suspect how many clusters there are in a set. There are several heuristics that can be applied to determine when a cluster should be split (when there appear to be two or more strong subclusters within it) or when two clusters need to be joined (there is not much difference between them). These heuristics can also be used to examine when channels might warrant being split or joined, by examining the features of those articles they contain.

- Dimensionality reduction. Dimensionality reductions (such as those achieved through LSI, PCA [principal-component-analysis], or SVD [singular-value-decomposition] techniques[26,27]) seek to map a given feature space to a space of much lower dimensionality through projection, where as much as possible of the important information is preserved. The hope is that operations such as finding the nearest neighbor or clustering can be performed much more efficiently in vector space of lower dimensionality. Unfortunately, the vectors in the reduced-dimensionality space tend to be opaque to a user; thus dimensionality reduction is at odds with the design goal of sharing everything with the user. In the ZWrap system, dimensionality reduction needs to be used carefully—never as the main feature in an expert.

User interface

The ZWrap system seeks to share its representation with the user at all times—it often uses less than mathematically optimal approaches in the interest of eliciting feedback from the user. The ZWrap system considers the user a resource that may be periodically employed and in general will have more "common sense" than the system does. ("Mathematically optimal" means doing the most with the information the system has. A technique that elicits additional user input may perform better than one that tries to make do with only existing information, since user input represents more information entering the system.)

To the extent possible, all information is stored internally in a form that is human-understandable. Having gone to this trouble, it only makes sense to then share as much of this information as possible with the user. This affords the possibility that the user will

notice when the system is "confused" and take steps to address it.

One goal in sharing information with the user is to fill in gaps in the user's understanding. If 15 cities in eastern Europe are mentioned in an article, a map might be used to present this information. If an unusual word appears, perhaps a dictionary definition would be useful. For individuals, a short biographic sketch can be provided. This type of augmentation requires a knowledge base and specialized experts, as in Elo's PLUM,[28] which uses augmentation to localize *FishWrap* articles about natural disasters.

One of the more frustrating features of many information-retrieval services is how hard it is to figure out *why* a particular document was selected for presentation. This is not just a trivial annoyance. Without understanding why a search engine produced an unwanted result, it is very difficult to modify an errant query to remedy the problem. The ZWrap system provides an explanation of how each article is selected for presentation.

Collections of articles are easier to skim if similar articles are grouped together. Traditional newspapers use sections such as "Sports" or "Living" to group their articles. An on-line newspaper can be more flexible; the ZWrap system allows users to define their own channels.

A World Wide Web page is not the only delivery mechanism for on-line news. There are also the printed page, pagers, electronic mail, telephones, instant messaging, audio alerts, LED (light-emitting diode) signs, etc. Restructuring of the presentation to use these various media is facilitated by the ZWrap internal structure.

It remains an open question how best to present an augmented news article. One truism about the user interface is that the more a system knows about both the news and the user, the better job it can do in presenting the users with the information they need. The ZWrap system addresses the news-representation half of this equation, but it must await an equally rich user-modeling system, e.g., DOPPELGÄNGER,[29] before its user interface develops further.

Implementation

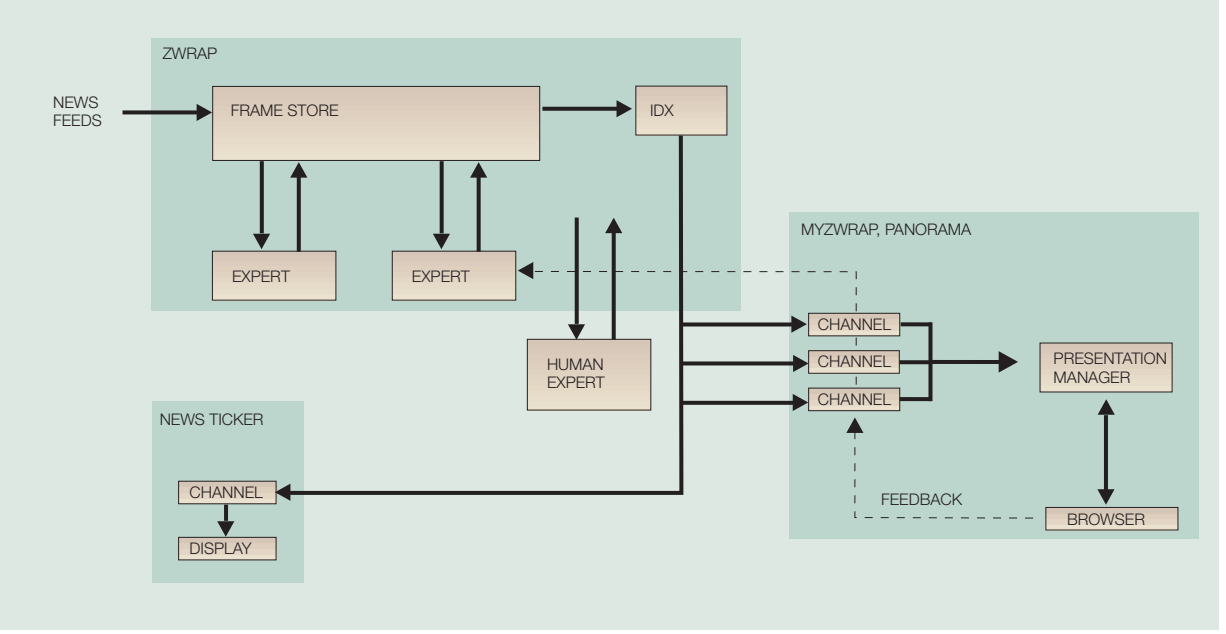
The ideas set out in the previous sections have been explored in two implementations. The first implementation, *MyZWrap*, is a general-purpose on-line news system developed and run at the MIT Media Laboratory. It obtains most of its news from the wire services (Associated Press World Stream**, Associated Press State 50**, Reuters, *The New York Times*), although it does get some from the Web (*The Onion*[30] as well as various sources of weather, comics, and sports). *MyZWrap* is designed to serve as a primary news source for individuals, providing news on a variety of general topics (similar in scope to sites such as www.cnn.com or www.usatoday.com).

Panorama, the second system, was designed and implemented at the IBM Almaden Research Center. It is a more focused on-line news system, designed to serve the needs of an electronics design engineer. Rather than employing wire services, *Panorama* obtains most of its news from the World Wide Web (www.cnn.com, www.usatoday.com, and company press pages) and Internet news and internal discussion sites. Since it is a more focused application, it utilizes domain-specific understanding (in the domain of the electronics industry) at the expense of a somewhat narrower understanding of the world at large.

In both of the projects, the same basic system was implemented (see Figure 4). The general flow of information is as follows. Articles enter the system through the news streams, having been acquired from a variety of sources. The articles are reformatted into frames and placed in a framestore. The experts examine the frames and augment them when appropriate. Statistical examination occurs in the background (trends that are observed are used in a number of ways, including the augmentation of the knowledge bases used by the experts). Searches are performed directly on the framestore and through various indexes that are computed. All of these features are exploited by the user interface to provide an augmented presentation.

Experts. *MyZWrap* and *Panorama* both use a large collection of experts to develop understanding. These experts connect to the framestore, request a frame, examine it, and add terminals to reflect their observations. As noted earlier, the experts talk to the framestore using the Dtype network protocol—they are written in whatever language is convenient. There is no constraint on how much an expert can "think." There is also no constraint on the use of human experts. For example, an editor might fine-tune the list of articles suggested by a channel server.

Figure 4 A block diagram of the ZWrap system and several prototype applications (*MyZWrap*, *Panorama*, and a news ticker)



Following is a list of some of the experts currently running:

- Structure. This is typically the first expert to run against an article. It uses a wide variety of heuristics to identify the various structural elements. For example, the word "by" followed by a proper noun is likely an indication of authorship if it appears in the first few lines of an article.
- Stemmer. The list of stemmed words is stored in the frame to facilitate word-based searches.
- Proper noun. A simple heuristic is used to identify proper nouns in an article.
- Noun/verb. Noun/verb pairs are identified in an article and included as features in the frame.
- Time spotter. References to time intervals, ages, and dates are identified in an article.[31] These references are converted to UNIX-style date/time.
- Place spotter. This expert identifies places mentioned in an article.
- Country spotter. Using a list of known countries drawn at run time from the CIA World On-Line Factbook,[32] this expert spots country names.
- Region spotter. This expert uses the country feature to identify those regions that are mentioned in an article, for example Middle East or South East Asia.
- People spotter. Using a list of known first names, this expert examines all the proper nouns and identifies persons who are mentioned in an article.
- Reading level. This expert makes use of an automated readability index to guess the "grade level" needed to comprehend an article.
- Media Lab faculty. A filter on the people feature is used to identify a Media Laboratory professor mentioned in an article.
- *FishWrap* topics. All of the *FishWrap* keyword topics are run and their matches written back into the frame.

This list is by no means exhaustive. Rather, it gives an idea of the span from the very general to the very specific, and illustrates how experts can work with each other. (An example of a frame augmented by experts is shown in Table 1).

Presentation and user interface. *MyZWrap* is a skewed project, with a disproportionately small amount of effort spent in exploring how articles are presented. Some issues have been examined in enough detail to merit mention:

- Top-level presentation. *MyZWrap* presents its information in channels. Each channel is focused on

Table 1 A frame augmented by the *MyZWrap* experts

Expert	Augmentation
Source	TEST
News Stream	TEST
Category	TEST
MD5	1be678600dc03fle25fc0797ae361762
Body	Nicholas Negroponte met with Vice President Al Gore during his visit to Boston last Wednesday
UNIX Time	934354800
Stemmer	{al, boston, dure, gore, hy, last, met, negropont, nichola, presid, to, vice, visit, wednesday, with}
Proper Nouns	{Nicholas Negroponte, Al Gore, Boston}
Noun/Verb	{Negroponte met}
Time Spotter	{Wednesday August 11, 1999 at 12:00 am EDT}
Place Spotter	{Boston}
Country Spotter	{United States of America}
Region Spotter	{North America}
People Spotter	{Nicholas Negroponte, Al Gore}
ARI Reading Level	3.87
Media Lab Spotter	{Nicholas Negroponte}
FishWrap Topic	{Politics, MIT, Media Lab}

a specific topic and the channel list is, in general, shared among users. *MyZWrap* places these channels in a three-column format. A graphical user interface is provided to allow simple channel selection as well as page-layout management.

- Searches vs repurposed information. *MyZWrap* is nonprescriptive regarding the implementation of channel servers. Not all channels perform searches on the news pool to generate their content. (Weather channels and comics acquire their news using other mechanisms, yet their presentation is wholly integrated with the other channels.)
- Channel creation. Repurposed news aside, the majority of *MyZWrap* channels are the results of searches. Since the system cannot anticipate all possible searches, some mechanism must be provided to enable channel creation. At the moment, the only "user friendly" channel-creation mechanism is an interface similar to AltaVista's that allows a search to be turned into a named channel. The "search explanation" feature in the ZWrap system allows existing channels to be fine-tuned or used as a basis for new channel creation.
- Channel analysis. *MyZWrap* provides some simple tools for channel analysis. The first is an examination of which features have recently appeared in articles that have been selected for a channel. By examining frequency of occurrence, the channel maintainer can identify active features and perhaps change the channel definition to account for them.
- New-feature alerts. Another category of tool is the "new feature" alert, of which WordWatch[33] is a good example. WordWatch mimics a "word-of-the-day" list. It creates its entries by examining the words that enter the system every day and finding "differences." These new words are filtered through a copy of the *Oxford English Dictionary* to rule out misspellings and the results are presented with definitions linked to the articles that triggered them. In general, bringing information to the attention of the user only when something new occurs minimizes the necessity for channel monitoring.

A few high-level engineering observations about the ZWrap approach to user interface are: (1) it allows complex real-time information-understanding and presentation architectures to be constructed out of simple pieces; (2) it allows the parts of the system to be distributed to arbitrary numbers of arbitrary types of machines for scalability; (3) it encourages development by allowing new components to be added without adversely impacting the existing system; (4) it encourages incremental improvement of existing components, since the system does not care how a component accomplishes its task; and (5) by involving users in channel creation and maintenance, the number of system administrators is kept to a minimum.

Results

There are formal methods of evaluation for many of the individual components of the ZWrap system. However, since the system is designed to readily incorporate new or improved components, their indi-

vidual evaluation is not particularly significant. It is more interesting to examine the system performance as a whole. The ZWrap system is an interactive information-retrieval system and, as such, it is difficult to construct a repeatable protocol for giving quantitative results. Side effects, such as users gaining familiarity with the task, differences between users, etc., create a system where an on-going interaction is difficult to characterize. One way to evaluate these systems is to have a large number of typical users work with the system and examine their interactions and opinions of the system. Such studies are expensive and often inconclusive. Another approach, becoming popular, is to release the system to the Internet and allow its merit to be determined by the number of hits. Some qualitative observations can, however, be made regarding how the ZWrap system fulfills its goal of multiple functionality.

Blackboard approach. Blackboard systems were initially developed to take advantage of their ease of development as well as opportunities for parallelism. Unfortunately, as noted, this approach fell somewhat into disfavor in the late 1980s due to the performance limitations resulting from the serialization of experts.

The ZWrap system is a validation of the blackboard architecture in the context of news. The system handles on the order of 10000 articles a day; this translates to approximately one gigabyte of text per month. In eight months, the system accumulated approximately 10 gigabytes in the total corpus of news. The ZWrap system can keep augmentations fully integrated to within approximately 20 minutes of the time that news enters the system. Finally, the current system is distributed across five desktop machines. That the system can maintain the approximate 20-minute performance on understanding with five machines working together argues that the multiple blackboard approach allows processing that is efficient enough to warrant its application to real systems.

Flexibility. The strength of a blackboard system is in the ease with which new components can be added to the system, and existing ones can be upgraded and improved upon. Two pieces of anecdotal evidence support this observation.

First, when developing the *Panorama* system, a full version of the system was implemented in roughly five days. This included the central blackboard, experts to post articles to the blackboard, a single augmentation expert (the stemmer) to test this portion, and the graphical user interface and article-selection structure. The speed with which enough of the system was developed for experimentation to begin was encouraging as it indicates that even on smaller projects, the overhead of including a blackboard approach should not be too burdensome.

The next observation is the ease with which new experts can be added to the system. While developing the technologies used by an expert to understand things may very well be a life's work, actually grafting them into the system is quite painless. A "food spotter" expert was created that, aside from problems in getting an account on the machine, took an afternoon to integrate. Likewise, a "color spotter" expert was implemented in less than an hour. This low overhead for including specialized understanding experts is heartening, as it encourages their development whenever a particular observation is needed to select articles correctly.

Scalability. The ability to add more hardware as needed is one feature that makes blackboard architectures attractive. Since all the components of the ZWrap system communicate over a network, adding more computational resources is accomplished by adding additional hosts to the network and reassigning services from one host to another. In general, changing hosts caused little impact.

At some point, it becomes impossible to realize performance improvements simply by segregating tasks to machines—the system becomes bound by the performance of the slowest expert running on a single machine. In these cases, it is usually possible to run more than one instance of an agent. Then the bottleneck moves to the database. Fortunately, extensive work has been done on allowing databases to handle large numbers of transactions, including multiple-node (e.g., Beowulf-type[34]) structures and serial-storage architectures.

These considerations aside, a five-machine cluster easily handles the loads discussed here and is sufficient to allow research on much larger dynamic real-time corpora than are traditionally contemplated.

Precognition. One constraint on any system that interacts with users is the need for short response times. This requirement limits the amount of computation that can be done while servicing a request and thus

Table 2 An experiment was performed to evaluate the impact of machine understanding informing statistics. The experimental procedure was to: (1) construct a dictionary of all terms in the training set; (2) construct a normalized vector for the entire training corpus; (3) construct a normalized vector for the training set; (4) identify the ten terms whose presence is most indicative of the training set as compared to the corpus norm; (5) identify the ten terms whose presence is most indicative of the corpus norm as compared to the training set; (6) use a projection onto these 20 dimensions to generate a cosine distance between each test article and both the "corpus-norm" point and the "training-norm" point, assigning each test article to the bin associated with whichever point is closest (i.e., select for topic or not). The results of an experiment that was tuned for a South American topic are shown. (South America is not a standard Reuters category. Country names and their mapping to continents were generated by the country- and region-spotter experts. A design feature of the ZWrap system is that specialized topics can be crafted when there is sufficient need.) It is not surprising that both precision and recall for the rules generated from augmented features were more than three times as accurate as for the rules generated without augmented features, since there is a nearly "perfect" feature that the system can use for classification. However, there is no reason not to add experts for a classification whenever possible.

Training Examples	With Augmentation		Without Augmentation	
	Precision	Recall	Precision	Recall
1	26.5	44.3	4.1	7.0
2	27.2	48.5	5.7	17.6
5	17.5	66.9	4.9	18.3
10	11.6	62.6	8.1	26.7
20	13.3	77.4	6.4	48.5
50	69.2	82.3	12.4	42.2
100	81.8	63.3	30.8	17.6
200	85.0	68.3	9.9	25.3
400	85.9	73.2	9.4	28.8

would seem to limit the complexity of the understanding that can be attempted.

The ZWrap system addresses this constraint by precognition, i.e., "thinking" about the articles before requests arrive. (Cognition here is meant to be the processing associated with human or machine augmentation. These "thoughts" are stored along with the article and can be quickly recalled as needed. Since much of the work is precomputed, complex operations can be executed without an adverse impact on response time.

Machine understanding in support of statistics. There is extensive literature on feature spotting as an adjunct to traditional information-retrieval methods.[35] The ZWrap system extends this, allowing domain-specific "spotters" to be added to the mix whenever it appears they will be helpful. New features can be added when required by the task at hand. (In the ZWrap system, all cases are treated as special cases if they are sufficiently important.) As is the case with most traditional information-retrieval methods, the ZWrap system can simply ignore those annotations (tokens) that are not helpful, although they may cause small amounts of confusion for limited corpora.

The Reuters-21578 dataset[36] was used in an experiment to evaluate the impact of feature augmentation on statistical classification as implemented in the ZWrap system. A typical set of activation-channel-selection rules for a ZWrap topic were created using both augmented and unaugmented articles as a training set and the results compared. The results of the experiment are shown in Table 2 and detailed in Gruhl.[2]

Statistics in support of machine understanding. The approach of using statistics on observations to develop rules for a knowledge base date back to at least Drescher,[37] where an expert made observations about the results of its actions in a simulated world and developed rules that it could later use to perform tasks. This is a goal of the ZWrap system but it is too ambitious a goal to implement in its entirety. One difficulty is that the ZWrap system cannot influence the news, but rather must make its observations based on the news. This limits the ability of the system to design experiments to fill the gaps in its understanding.

Rather than abandoning this approach altogether, the ZWrap system seeks to identify potential causality and brings this to the attention of a person or expert with the broader understanding to "fill in the blanks" and decide whether the observation is indeed valid.

Table 3 A data-mining experiment was performed to evaluate the impact of statistical classification on knowledge-base construction as implemented in the ZWrap system. Association-rule data mining was applied to proper-noun features and those features with high co-occurrence rates. (Both human knowledge engineers and deep machine-understanding processes are treated as expensive, limited resources, to be used sparingly.) By looking for implications among high-occurrence features, the system seeks to focus development on those areas that will have substantial impact. The experiment was performed on proper nouns occurring in a two-week period of news in February 1999. The association rules that were identified had a minimum support of 20 articles and a confidence of at least 50 percent.

Term A	Implies	Term B	Support	Confidence
West Bank	→	Israel	277	0.711
West Bank	→	Israeli	277	0.632
West Bank	→	Palestinian	277	0.610
West Bank	→	Palestinians	277	0.560

Table 4 Associations between "Richard Butler" and other proper nouns over a two-week period in February 1999. This is the complete rule list generated with a support requirement of 20 articles and a confidence requirement of 50 percent.

Richard Butler (28 examples)	Associations (% of co-occurrences)
Iraq	89
Security Council	64
Special Commission	53
British	6
Iraqi	82
UNSCOM	54

The example in Table 3 illustrates how a candidate geographic rule is generated. Other associations discovered in the experiment include biographic and short-duration rules (see Table 4). The rules are presented to a human critic in order to assess the suitability for their inclusion in the knowledge base. Casting knowledge engineers in the role of rule critics, rather than rule creators, lightens their load.

Conclusion

This work was motivated by four observations: (1) the general lack of an efficient, flexible way to deal with large, evolving corpora in a nontrivial manner and the general notion that large corpora require simpler techniques than small corpora; (2) the perceived hard division between machine-understanding approaches to information retrieval and those developed from a purely statistical basis; (3) the tendency of systems that perform any understanding of their text to quickly move to representations that are opaque to human comprehension; and (4) the extent to which information retrieval systems fail to share any of their understanding with their human users. The consequence is that the user has little opportunity to enhance the development of meaning, although we have found that representations that allow discretionary use of human judgment are of great value. From the ZWrap architecture, developed to address these issues, two test applications were constructed.

The architecture overcomes the traditional deficiencies of the blackboard architecture and uses this approach to build rich representations of large, dynamic corpora. In doing so, the architecture provides a framework in which a "society of agents" approach can be scaled up and applied to large text-understanding problems. Experts are allowed to interact in a controlled way through the blackboard and can be distributed over available computational resources. In addition, the architecture provides a lightweight, reusable structure.

The architecture allows computationally intensive investigation of articles to be performed ahead of time, and the resultant structures stored. This allows more in-depth examination of articles at search time without the need for the user to wait for the results.

Linking statistical and machine-understanding systems, the ZWrap system demonstrates the suitability of frame-type techniques for very large (i.e., millions of documents) collections of information. It exemplifies the use of data-mining statistical methods to assist in the creation of knowledge bases for machine-understanding systems and it exemplifies the use of machine-understanding feature identification to assist in statistical clustering. The ZWrap system also demonstrates that such a system can maintain a human-readable internal representation and yet still perform efficiently.

Finally, the task of an editor is to assess both the news and its context. The augmented-frame model used in the ZWrap system offers some help with this task by providing a structure for the sharing of information that it develops with the user. But, as the ZWrap system currently uses only a cursory model of the user, editorial questions such as "What would the reader like to know about?" are difficult to answer with any precision. Still, the ZWrap system fosters a collaboration between the system and the user, and like all collaborations, the better the communication, the better it works.

A distributed but structured approach to on-line news brings the possibility of more participants in the editorial process, each adding value to the whole. This might result in a reversal of roles—the reader becomes the editor. It will certainly result in a new relationship between readers and editors.

Acknowledgments

The authors would like to thank Charles Coffing and Jeremy Braun for their help in implementing ZWrap. We would also like to acknowledge Ken Haase, Klee Dienes, Nathan Abramson, Pascal Chesnais, the *FishWrap* development team, and the current and past members of the Electronic Publishing Group. This work was supported in part by IBM and the MIT Media Laboratory's News in the Future research consortium.

Cited references

1. M. Massey and W. Bender, "Salient Stills: Process and Practice," *IBM Systems Journal* **35,** Nos. 3&4, 557–574 (1996).
2. D. Gruhl, *The Search for Meaning in Large Text Databases*, Ph.D. thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, MA (December 1999).
3. J. Montbriand, Extending and Controlling Sherlock, Technote 1141, http://developer.apple.com/technotes/tn/tn1141.html (October 1998).
4. A. Lippman and R. Kermode, "Media Banks: Entertainment and the Internet," *IBM Systems Journal* **35,** Nos. 3&4, 272–291 (1996).
5. A. Newell, "Some Problems of the Basic Organization in Problem-Solving Programs," *Proceedings of the Second Conference on Self-Organizing Systems*, M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, Editors, Spartan Books, Washington, DC (1962), pp. 393–423.
6. R. S. Engelmore, A. J. Morgan, and H. P. Nii, "Introduction," *Blackboard Systems*, Addison-Wesley Publishing Company, Reading, MA (1989), pp. 1–22.
7. M. Minsky, *The Society of Mind*, Simon & Schuster, Inc., New York (1988).
8. R. S. Engelmore and A. J. Morgan, "Conclusion," *Blackboard Systems*, Addison-Wesley Publishing Company, Reading, MA (1989), pp. 561–574.
9. L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The Hearsay Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," *Blackboard Systems*, R. S. Engelmore and A. J. Morgan, Editors, Addison-Wesley Publishing Company, Reading, MA (1989), pp. 31–86.
10. H. P. Nii, E. A. Feigenbaum, J. J. Anton, and A. J. Rockmore, "Signal-to-Symbol Transformation: HASP/SIAP Case Study," *Blackboard Systems*, R. S. Engelmore and A. J. Morgan, Editors, Addison-Wesley Publishing Company, Reading, MA (1989), pp. 135–157.
11. D. B. Lenat and R. V. Guha, *Building Large Knowledge Based Systems: Representation and Interference in the Cyc Project*, Addison-Wesley Publishing Company, Reading, MA (1990).
12. N. Carver and V. Lesser, "The Evolution of Blackboard Control Architectures," *Expert Systems with Applications* **7,** No. 1, 1–30 (1994).
13. M. Minsky, "A Framework for Representing Knowledge," *Technical Report MIT AI Laboratory Memo 306*, Massachusetts Institute of Technology Artificial Intelligence Laboratory, Cambridge, MA (June 1974).
14. K. Haase, "FramerD: Representing Knowledge in the Large," *IBM Systems Journal* **35,** Nos. 3&4, 381–397 (1996).
15. N. Abramson and W. Bender, "Context-Sensitive Multimedia," *Proceedings of the SPIE: Enabling Technologies for High-Bandwidth Applications* **1785,** Boston, MA (September 10–11, 1992), pp. 122–132.
16. K. Dienes, *Information Architectures for Personalized Multimedia*, M.S. thesis, MIT Program in Media Arts and Sciences, Cambridge, MA (1995).
17. P. Chesnais, J. Sheena, and M. Mucklo, "The FishWrap Personalized News System," *Proceedings of the Second International Workshop on Community Networking*, IEEE, Princeton, NJ (June 20–22, 1995), pp. 275–282.
18. See http://www.altavista.com.
19. G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," *Journal of the American Society for Information Science* **41,** No. 4, 288–297 (1990).
20. K. Ng, "Information Fusion for Spoken Document Retrieval," *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey (June 5–9, 2000).
21. A. Lippman and W. Bender, "News and Movies in the 50 Megabit Living Room," *Proceedings IEEE/IECE Global Telecommunications Conference*, Tokyo (November 15–19, 1987), pp. 1976–1981.
22. See http://www.pointcast.com/.
23. See http://www.excite.com.
24. H. Turtle and W. B. Croft, "Inference Networks for Document Retrieval," *Proceedings of the Thirteenth International Conference on Research and Development in Information Retrieval*, Brussels, Belgium (September 5–7, 1990), pp. 1–24.
25. P. Heckbert, "Color Image Quantization for Frame Buffer Display," *Computer Graphics* **16,** No. 3, 297–304 (1981).
26. G. H. Golub and C. F. Van Loan, *Matrix Computation*, *Third Edition*, The Johns Hopkins University Press, Baltimore, MD (1996).

27. P. A. Derijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall International (1982).
28. W. Bender, P. Chesnais, S. Elo, A. Shaw, and M. Shaw, "Enriching Communities: Harbingers of News in the Future," *IBM Systems Journal* **35,** Nos. 3&4, 369–380 (1996).
29. J. Orwant, "For Want of a Bit, the User Was Lost: Cheap User Modeling," *IBM Systems Journal* **35,** Nos. 3&4, 398–416 (1996).
30. See http://www.theonion.com.
31. D. B. Koen and W. Bender, "Time Frames: Temporal Augmentation of the News," *IBM Systems Journal* **39,** Nos. 3&4, 597–616 (2000, this issue).
32. Central Intelligence Agency, *The World Fact Book 1999*, http://www.odci.gov/cia/publications/factbook/index.html (1999).
33. See http://nif.www.media.mit.edu/WordWatch.
34. T. Sterling, D. Becker, D. Savarese, J. E. Dorband, U. A. Ranawak, and C. V. Packer, "Beowulf: A Parallel Workstation for Scientific Computations," *Proceedings, International Conference on Parallel Processing* **1,** CRC Press LLC, Boca Raton, FL (1995), pp. 11–14.
35. E. Riloff and W. G. Lehnert, "Information Extraction as a Basis for High-Precision Text Classification," *ACM Transactions on Information Systems* **12,** No. 3, 296–333 (1994).
36. Reuters-21578 dataset, http://www.research.att.com/~lewis/reuters21578.html.
37. G. L. Drescher, *Made-up Minds*, MIT Press, Cambridge, MA (1991).

Accepted for publication May 1, 2000.

Daniel Gruhl *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099 (electronic mail: dgruhl@almaden.ibm.com)*. Dr. Gruhl is a research staff member at the IBM Almaden Research Center, where he is a member of the Exploratory Database Systems group. He received his doctorate degree in electrical engineering from MIT in 2000; his research was done at the MIT Media Laboratory.

Walter Bender *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: walter@media.mit.edu)*. Mr. Bender is a senior scientist at the MIT Media Laboratory and principal investigator of the laboratory's News in the Future consortium. He received the B.A. degree from Harvard University in 1977 and joined the Architecture Machine Group at MIT in 1978. He received the M.S. degree from MIT in 1980. Mr. Bender is a founding member of the Media Laboratory.