ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220354298

A new structure for news editing

Article *in* Ibm Systems Journal · January 2000

DOI: 10.1147/sj.393.0569 · Source: DBLP

CITATIONS
6

READS
4,357

2 authors:

Daniel Gruhl
IBM
70 PUBLICATIONS **6,503** CITATIONS

SEE PROFILE

Walter Bender
Massachusetts Institute of Technology
67 PUBLICATIONS **4,197** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Movie Maps View project

All content following this page was uploaded by Walter Bender on 31 May 2014.

The user has requested enhancement of the downloaded file.

A New Structure for News Editing

by D. Gruhl
W. Bender

Ideally a computational approach could alleviate 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 towards 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 on-line news. These representations are composed by a collection of agents that examines news articles in the database, looking at both the text itself and the annotations placed by other agents. These agents 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 leverages article augmentation.

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 impact them, but at the same time 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 Monica Lewinsky article a week might be interesting. Fifty might not be. A person’s source of news needs to express “what’s new” rather than just “what’s happened.” However, if someone has relatives in Pakistan, then *every* article about a revolution that occurs there may be of interest.

The average person has many interests which compete for their time. The time one can spend on the news each day is more or less fixed, and decisions must be made what to present, in what order, and in what way. It requires understanding on the editor's part as to 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 out there, 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 to keep and what to discard?

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, what their strengths are, as well as their weaknesses.

"In today's Journal of the American Medical Association..."—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 TV weather report in the morning, the newspaper on the bus ride to work, the radio for the outcome of the afternoon Sox 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 articles are being selected from all of them, then they 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; as well as simple hard numbers like newsstand sales when a particular headline is run. This feedback allows the editor to better serve the needs of their community. Note that none of this feedback is actively requested 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 “wait.” 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 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 what was too restrictive in the request. Other times they return far too many results, swamping the user with a plethora of information they then must wade through and decided on for themselves. Neither of these alternatives is particularly attractive to an end user. Little wonder then that most people 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 don’t 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 they trust the editors will provide the information they need. An on-line newspaper will need to be able to provide users with some reasonable starting point, even if they themselves don’t know what that is.

I want what he’s got!—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, people lack a common reference point with which to engage in discussion.

News as a service. Ideally a computational approach could alleviate 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 towards improving both the efficiency and utility of on-line news. This structure reflects a reorganizing of on-line news distribution around a services model—including services for: (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; (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 in and deployed with relative autonomy.

The ZWrap system. This paper examines an approach by which a computer system, ZWrap[1], can develop rich structure of 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, most 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.) ZWrap considers what technologies are needed to keep this context current in the face of a changing external world. It examines ways in which non-information retrieval experts can share their understandings with the system, and act as a source of common sense for it. The goal is to develop a system that can assist, simplify and automated the types of editorial decisions that a human editor must face and resolve, doing so in a way that is amenable to use in an on-line news environment. The result is a system that is easy to assemble, easy to maintain, easy to improve on, and performs the task of developing and discovering meaning in a reasonably efficient and interesting manner.

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 “pre-cognition,” modular, 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, 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. Roughly speaking, a blackboard system is one that employs a collection of experts that independently look at a “blackboard” and judge when they have something useful to add[2].

In a blackboard system, the problem being considered is written on a blackboard. A group of experts (or agents) sits in front of this blackboard, examining everything on it. There is one piece of “chalk” and when an expert has something to contribute to the understanding of a problem, they take the chalk and write their observation 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 they may make use of observations written on the blackboard by other experts.

Creating a piece of code for an expert is fairly simple. It needs to be able to read the blackboard, grab 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 agent 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 parallelizing the “cognition” process.

From a theoretical standpoint, this architecture allows for the development of a “society of agents,” as suggested by Minsky[3], with a number of very specialized experts contributing their observations to a problem in hopes of finding some kind of understanding about it. Each of these experts can “evolve” independently, new ones can be added at any time, and ones that are found to be less useful dropped in favor of those more so.

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. If agent B needs to look at agent A’s observations, then until agent A makes those observations, agent B can do except 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 a fight 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 1A). It has been observed that blackboard systems perform less efficiently than many alternative architectures.

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[4], sonar contact analysis[5], and fighter aircraft intercept evaluation[6], 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 the weakness of blackboard systems.

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 agents 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 agents that depend on the work of others can just visit blackboards that already have the appropriate colored marks on them. In short, most all of the problems with the original blackboard architecture either do not arise or can be avoided in the case of news systems (See Figure 1B).

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. Agents watch a variety of news sources. Whenever a article arrives, a new blackboard is created, the article is posted, and the blackboard is then visited by a number of experts.

In the ZWrap system, a group of experts examines the articles for purely structural elements. They extract the headline, the dateline, the author, the body text, the time the article was posted, etc. and place this information on the blackboard. These experts mitigate the problem that different news sources provide news having different structure. These agents 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 agents.

A second group of agents performs information extraction, parsing, and evaluation on the uniform article elements extracted above. They 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 of the article.

A third group of agents uses the derived features to perform high-level augmentation. For example, the people spotter looks at the list of proper nouns and decides (through a simple heuristic) which proper nouns are the names of people (See Figure 2). 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 crowd are agents that use even these derived features to produce more features. For example, the Media Lab agent 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.

A word about dependency—in Figure 2 the Person expert could do no work until the Proper Noun expert had visited the article. This could lead to inefficiency if the Person expert kept checking blackboards to see if they had been processed. ZWrap deals with this through the use of a dependency scratch space. When an expert is finished with a set of blackboards, it notes this in the dependency scratch space. This allows the other experts to know when it is worth checking the blackboards.

Data Representation

The use of a blackboard system simplifies the question of data representation as 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[7] as the data storage medium. A frame is a collection of key/value pairs, known as terminals, that describe a particular object or concept.

Frames have a number of advantages, not the least of which are 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 need in a frame is important since it allows the addition of arbitrary experts without the need to modify any other agents in the system.

A feature borrowed from FramerD[8], 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) to refer to every frame. This 64-bit number allows the frame to be referred to in a succinct, unambiguous manner. Since this OIDs are never recycled, they can be used as a "pointer" to an article and it is assured that the article 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 terminals to the frame. A mechanism exists to allow for locking of a blackboard when an expert is looking at it (inherited from FramerD).

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 ZWrap uses a framestore to represent its blackboards, the question arises as to how will the agents 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.

Light-weight framestore access—The primary concern of someone coding an expert should not be how does that agent 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. If an expert connects to the service, then they can allow the service provider to worry about issues of concurrent access—lightening the load on the expert and those authoring 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 agents 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 speak the framestore protocol can operate as an expert. Thus, experts can be written in whatever language is appropriate for the processing they seek to perform and run on whatever hardware or operating system that best facilitates their execution.

All that is needed to employ 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 towards the task at hand, not 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 old experts, e.g., when, for efficiency, the ability to grab “blocks” of frames is added.

Expressiveness—This is key for experts that might be writing 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 an ASCII 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.

It is fortunate that a protocol meeting almost all of these criteria existed at the inception of the ZWrap project in the form of Dtypes. Dtypes were originally developed by Abramson[9], later extended by Dienes[10] and subsequently formalized and extended by Haase and Gruhl. The resulting protocol meets all of the above criteria and serves as the underlying communication system for ZWrap.

Expert Distribution. Any program that can connect over the network to the server and send the above commands is a potential “expert” in the ZWrap system. Experts can run on any machine. At the moment, ZWrap uses 4–6 machines to augment and present the news. These machines have been (over the course of this investigation) RS6000s, Suns, Macintoshes, Next Machines, Linux Intel machines, Windows NT machines and Windows 95/8 machines. Since the Dtype library is so simple to implement, versions exist in Java, Perl, C, C++, Scheme and Lisp, allowing experts to be implemented in whatever language is most practical for the task they will undertake.

The use of the Dtype 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 mid-range commercial computers allows the use of many computationally-expensive experts without the need to resort to expensive special-purpose super-computing solutions.

The distribution of agents over a network allows for the simultaneous pursuit of many 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 ZWrap 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. 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. ZWrap implements this type of search using the operators **AND**, **OR**, and **NAND**. However, ZWrap operates on terminals when searching, not words. Since the set of stemmed words is written back into the frame, the above search (in ZWrap) is implemented as

((STEMMED_WORD . “Bill”) OR (STEMMED_WORD . “Hillary”)) AND (STEMMED_WORD . “Clinton”).

But **STEMMED_WORD** 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. A ZWrap expert has been written that spots types of food and notes it under the **FOOD** terminal. A number of experts are inherited from *Fishwrap*[11] (an on-line news system that is a precursor to ZWrap), one of which spots morbid news stories, e.g., articles where death, serious injury, or grievous harm occur. A geography agent places information on continents implicitly mention 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 agents are employed and some clever authoring of search expressions, reasonably complex concepts can be expressed with just Booleans 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 are their typical values. Second, there is no concept of how “on topic” an article is. It either matches the Boolean or it does not. There is very little in the way of “hints” that can be passed along to, for example, the display engine to help decide how the articles should be presented.

Weighted searches. A weighted search, such as AltaVista[12], 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 have Bill and Clinton in them but do not have 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 ZWrap 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 or detracts from an article's score for selection. For example

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.

For readability, words are 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 example suggests that relevance feedback might be a useful approach for designing searches: The user performs an initial search to identify articles similar to the ones they are seeking; The system looks at for similarities between these articles and uses this as a search criteria; The user examines the results of this new search and identifies articles that seem most relevant; This process iterates until the user has found articles of the class they are looking for.

Relevance-feedback-type searches need not be explicit. If the system can observe the user interactions and infer something about which articles were of interest, then this approach can be used to refine news channels without explicit formulation of rankings.

News channels. ZWrap borrows the concept of channels for news presentation from *NewsPeek*[13], PointCast[14], and MyExcite[15]. 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 ZWrap, any search performed by the user is a candidate for repurposing into a channel. This allows a user to employ any search skills they might have towards the task of automating the editing of their 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 to an agent. 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 agent.

Statistics

ZWrap exploits how the frames relate to each other through a variety of statistical examinations of the corpus, looking for patterns and trends that develop between high-level features.

Developing good searches by hand is effortful. This is confounded by the tendency for topics to “drift” over time. As explained in the previous section, ZWrap 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 agents, 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 the list of 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.

Since features carry more information than words, techniques that work well on word sets work even better on augmented frames. In ZWrap, by the time an article is to be examined statistically, it has additional features that have been spotted, computed, or otherwise added to the frame; it is possible to use a “feature document matrix.” It is this matrix that is used by agents to provide context for individual articles and to find associations and differences between multiple articles.

Statistical techniques. Each article in ZWrap is represented as a vector of features, where a_i is the feature vector for article *i* and $a_{i(n)}$ the *n*th 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, ZWrap uses Boolean features. This means that feature vectors are filled with only ones and zeros, representing the presence or absence of a feature in a given article.

ZWrap 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. In ZWrap the user is considered a resource, which 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 works in such a way as to elicit additional user input may out-perform one that tries to make do with only the information it starts with, since user input represents more information entering the system.)

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.

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 $\cos(A \cdot \bar{A})$ is then a measure of how "distant" a particular article is from what is typical for the channel. The set of $\bar{A}$s 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 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, looking at what features occur together frequently in the same article. Association-rule data mining 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 clusters. K-means is run on a set of LSI-dimensionality-reduced vectors generated from the stemmed-word document vectors through singular-value decomposition. ZWrap gives the user an indication of why a cluster has been created by revealing those high-level features that are in common between 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 ass the system to expand on an article to indicate that they are interested in its associated cluster.

Cluster management—Most clustering algorithm 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 cluster there are in a set. There are several heuristics that can be applied to determine when a cluster should be split (when there appears to be two or more strong subcluster 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 examine the features of those articles they contain.

Dimensionality reduction—Dimensionality reductions (such as those achieved through LSI-, PCA- and/or SVD-type techniques[16, 17]) 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 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, thus dimensionality reduction is at odds with the design goal of sharing everything with the user. In ZWrap, dimensionality reduction needs to be used carefully—never as the main feature in an expert.

User Interface

The task of an editor is to assess both the news and its context. The augmented-frame model used in ZWrap offers some help with this task, but, as ZWrap currently deploys only the most cursory model of the user, editorial questions such as “What would the reader like to know about?” are difficult to answer with any precision. With this reservation in mind, however, it is worth examining some of the design principles that went into developing the ZWrap user interface as it currently exists. In short, the philosophy guiding the design of the user interface is that understanding is a collaboration between the system and the user, and like all collaborations, the better the communication, the better they work.

Share everything—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.

Article augmentation—Information is also shared with the user with the goal of filling in gaps in their understanding. If 15 cities in eastern Europe are mentioned in an article, a map might be used to present this information. If the occasional odd word appears in an article, perhaps a dictionary definition would be useful. With people, a short biographic sketch. This type of augmentation requires a knowledge base and specialized agents, as per Elo’s PLUM[18], which uses augmentation to localize *FishWrap* articles about natural disasters.

What was it thinking?—One of the more frustrating feature 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. ZWrap provides an explanation of how each article is selected for presentation.

Channel structure—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; ZWrap allows users to define their own channels.

Alternative delivery mechanisms—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 signs, etc. Restructuring of the presentation due the differing nature of 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 presenting the users with the information they need. ZWrap addresses the news-representation half of this equation, but it must await an equally rich user-modeling system, e.g., Doppelgänger[19], 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, *New York Times*), although it does get some from the web (*The Onion* 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 a site 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 3). The general flow of information is as follows: Articles enters 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. Here is a list of some of the agents currently running:

Structure—Typically the first expert to run. It uses a wide variety of heuristics to identify the various "structural" elements of an article. 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[20]. 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 runtime from the CIA World On-Line Factbook[21], 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.

Continent spotter—This expert maps country to continent.

People spotter—Using a list of known first names, this expert examines all the proper nouns and identifies those people who are mentioned in an article.

Reading level—The expert makes use of an automated readability index to guess the "grade level" needed to comprehend of an article.

Media Lab faculty—A filter on the people feature is used to identify when a Media Laboratory professor is 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 agents can work with each other.

Presentation and user interface. *MyZWrap* is defiantly a skewed project, with a disproportionately small amount of effort having been dedicated to 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 a specific topic and the channel list is, in general, shared between users. *MyZWra*p places these channels in a 3-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 agnostic regarding the implementation of channel servers. Not all channels perform searches on the newspool to generate their content. (Weather channels and comics acquire their news using non-search 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 AltaVista-like interface that allows a search to be turned into a named channel. The "search explanation" feature in ZWrap enables existing channels to be fine-tuned or used as a basis of 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 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 find "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 for the construction of complex real-time information-understanding and presentation architectures out of simple pieces; (2) It allows for scalability by distributing the parts of the system to arbitrary numbers of arbitrary types of machines; (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 agnostic about 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

The ZWrap system as a whole 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.

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 has fallen into a certain disfavor in the late eighties due to the performance limitations resulting from the serialization of agents.

ZWrap is a validation of the blackboard architecture: ZWrap handles on the order of 10 000 articles a day; This translates to approximately one gigabyte of text per month; At the time of the authoring of this document, ZWrap has been running (and gathering news) internal to the Media Laboratory for about eight months; This translates to around 10 gigabytes in the total corpus of news; The system can keep augmentations fully integrated to within approximately 20 minutes of when news enters the system; The system currently runs distributed across five Intel-type machines. That the system can maintain the approximate 20 minute performance on understanding with the five machines working together argues that the multiple blackboard approach allows for efficient enough processing of text to warrant its application to real systems.

Flexibility. The strength of a blackboard system is in the ease at 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 "through" version of the system was implemented in roughly five days. This included the central blackboard, agents to post articles to the blackboard, a single augmentation agent (the stemmer) to test this portion, and the graphical user interface and article-selection structure. The speed of development of enough of the system to start performing experiments 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 to which new agents can be added to the system. While developing the technologies used by an agent to understand things may very well be a life's work, actually grafting them into the system is quite painless. A "food spotter" agent was created that, aside from issues of getting an account on the machine took them an afternoon to integrate. Likewise, a "color spotter" agent was implemented in less than an hour. This low overhead of including specialized understanding agents is heartening, as it encourages development of them 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 which services are performed on which hosts. In general, there is little impact in changing the host with which a component communicates for a particular service.

At some point, it becomes impossible to realizing performance improvements simply by segregating tasks to machines—the system becomes bound by the performance of the slowest agent running on a machine by itself. The bottleneck moves to the database. Fortunately, extensive work has been done on allow databases to handle large numbers of transactions, including multiple-node and serial-storage architectures.

ZWrap takes an even simpler approach. A hashing function is applied to the frame OID and used to identify the server that holds a particular frame. The hash is used to partition the framestore into an arbitrary numbers of parts. Coupled with a fast-switching network with dedicated lines of communication between machines, there is no reason to believe that this structure could not grow to on the order of a hundred machines. By using Beowulf-type[22] structures, the system can quickly grow to thousands of machines, as each of the nodes can be replaced with Beowulf clusters.

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

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

ZWrap addresses this constraint by pre-cognition, i.e., “thinking” about the articles before requests arrive. 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 informing statistics. There is a fair amount of literature on feature spotting as an adjunct to traditional information-retrieval methods[23]. ZWrap employs an extension to this of allowing domain specific “spotters” to be added to the mix whenever it appears they will be helpful. Most traditional information-retrieval methods can simply ignore those annotations (tokens) which are not helpful, although they may cause small amounts of confusion with limited corpora.

The Reuters-21578 dataset[6] was used in an experiment to evaluate the impact of feature augmentation on statistical classifying as implement 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 to compare the results.

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 train set; (4) Identify the 10 terms whose presence is most indicative of the training set as compared to the corpus norm; (5) Identify the 10 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 the experiment that was tuned for a South American topic are shown in Table 2. (County names and their mapping to continents were generated by the country-, region-, and continent-spotter experts.) It is not surprising that both precision and recall for the rules generated from augmented features were more than 3 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.

Statistics informing machine understanding. The approach of using statistics on observations to develop rules for a knowledge base date back to at least Drescher[24], where an agent 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 ZWrap but it is too ambitious a goal to implement in its entirely. 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 all together, ZWrap seeks to identify where it suspects that there is a causality and brings this to the attention of a person or expert that has the broader understanding to “fill in the blanks” and decide whether the observation is indeed valid.

A data-mining experiment was performed to evaluate the impact of statistical classification on knowledge-base construction as implement in the ZWrap system. Association-rule data mining[25] was applied to proper-noun features and examining 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%. The example shown in Table 3 illustrates a candidate geographic rule is generated. Other associations exposed in experiment include biographic and short-duration rules. 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 in which to deal with large, evolving corpora in a non-trivial 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. This has the consequence that the user has little opportunity to enhance the development of meaning—indeed, we have found that representations that facilitate discretionary use of human judgement are of great value. The ZWrap architecture was developed that addresses these issues from which two test applications were constructed.

The architecture addresses the traditional deficiencies of the blackboard architecture and thus utilizes 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. Agents 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 light-weight, 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 (using the precached features) without the need for the user to wait for the results.

Regarding a linkage between statistical and machine-understanding systems, ZWrap 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-class statistical methods to assist in the creation of knowledge bases for machine-understanding systems and it exemplifies the use of machine-understanding-class feature identification to assist in statistical clustering. ZWrap also demonstrates that such a system can maintain a human-readable internal representation and yet still perform efficiently.

Finally, the architecture provides a structure for the sharing information that it develops with the user, opening the door to a wide class of applications that leverage article augmentation. At the same time, the architecture provides a structure for the integration back into the system of any information or understanding provided by the user.

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

References

1. D. Gruhl, *The Search for Meaning in Large Text Databases*, Ph.D. Thesis, MIT EECS, December (1999).
2. R.S. Engelmore, A.J. Morgan, and H.P. Nii, "Introduction," *Blackboard Systems*, Addison-Wesley Pub. Ltd. (1988).
3. M. Minsky, *The Society of Mind*, Simon & Schuster (1988).
4. 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*, Addison-Wesley Pub. Ltd. (1988).
5. H.P. Nii, E.A. Feigenbaum, J.J. Anton, and A.J. Rockmore, "Signal-to-symbol transformation: HASP/SIAP case study," *Blackboard Systems*, Addison-Wesley Pub. Ltd. (1988).
6. D.B. Lenat and R.V. Guha, *Building Large Knowledge Based Systems*, Addison-Wesley Pub. Ltd. (1989).
7. M. Minsky, "A framework for representing knowledge," *Technical Report MIT AI Laboratory Memo 306*, Massachusetts Institute of Technology Artificial Intelligence Laboratory (June 1974).
8. K. Haase, "FramerD," *IBM Systems Journal* **38**, nos. 3&4, pp. (1996).
9. N. Abramson and W. Bender, "Context-sensitive multimedia," *Proceedings of the SPIE* **1785** (1992).
10. K. Dienes, *Information Architectures for Personalized Multimedia*, MS Thesis, Massachusetts Institute of Technology Program in Media Arts & Sciences (1995).
11. P. Chesnais, J. Sheena, and M. Mucklo, "The FishWrap personalized news system," *Proceedings IEEE Second International Workshop on Community Networking* (1995).
12. Compaq Corporation, *The AltaVista Search Engine*, http://www.altavista.com
13. A. Lippman, W. Bender, "News and movies in the 50 megabit living room," *IEEE Globecom*, Tokyo (1987).
14. *The Pointcast Network*, http://www.pointcast.com/
15. Excite, Inc., http://www.excite.com
16. G.H. Golub and C.F. Van Loan, *Matrix Computation, Third Edition*, The Johns Hopkins University Press (1996).
17. P.A. Derijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall International (1982).
18. W. Bender, P. Chesnais, S. Elo, A. Shaw, M. Shaw, "Harbingers of news in the future," *IBM Systems Journal* **38**, nos. 3&4, pp. (1996).
19. J. Orwant, "For Want of a Bit, the User was Lost," *IBM Systems Journal* **38**, nos. 3&4, pp. (1996).

20. D.B. Koen and W. Bender, “Time frames: Temporal augmentation of the news,” submitted to the *IBM Systems Journal* (October 1999).

21. Central Intelligence Agency, *The World Fact Book 1999*, http://www.odci.gov/cia/publications/factbook/index.html (1999).

22. D. Becker, T. Sterling, 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* (1995).

23. E. Riloff and W.G. Lehnert, “Information extraction as a basis for high-precision text classification,” *ACM Transactions on Information Systems* (1994).

24. G.L. Drescher, *Made-up Minds*, The MIT Press (1991).

25. H. Turtle and W.B. Croft, “Inference networks for document retrieval,” In *13th International Conference on Research and Development in Information Retrieval* (September 1990).

Figures and tables

Figure 1 (A) On the left, Lots of experts, all standing around the same blackboard. They all look at the blackboard, and when one has something to contribute, they grab the chalk (if it is available) and write their thoughts down. All communication is done via the blackboard, and only one expert is allowed to write at a time. On the right as you might imagine, this can lead to a lot of frustrated agents! (B) With a group of blackboards, experts can wander around, writing on whichever blackboard is free.

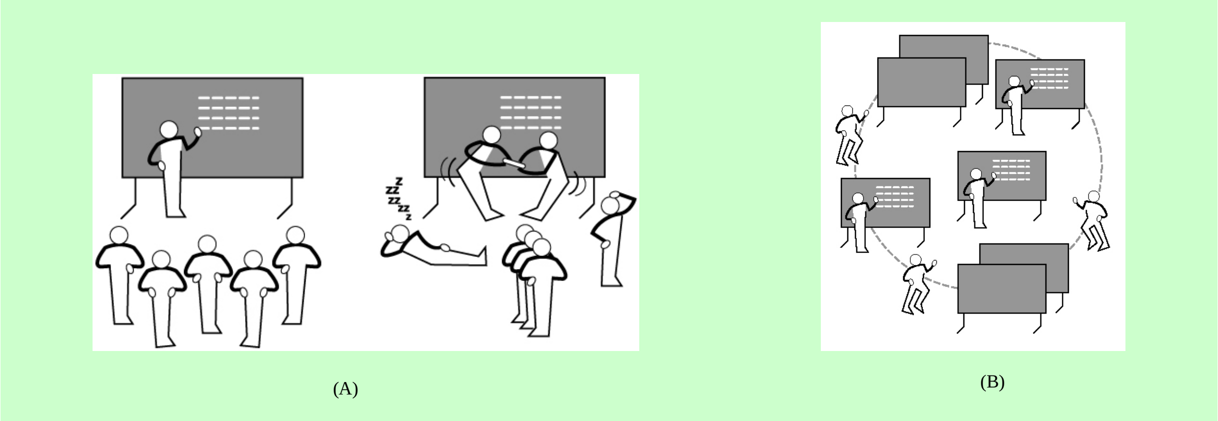

Figure 2 The proper-noun expert adds a list of the proper nouns found in an article. The person expert examines this list to identify the names of people and writes them into the article.

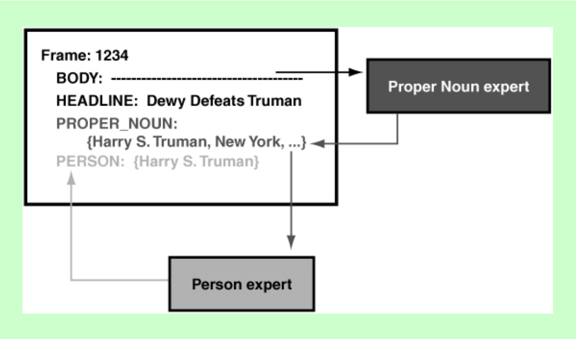
Frame: 1234
BODY: --
HEADLINE: Dewy Defeats Truman
PROPER_NOUN:
{Harry S. Truman, New York, ...}
PERSON: {Harry S. Truman}
Proper Noun expert
Person expert

Figure 3 Overall system diagram for ZWrap

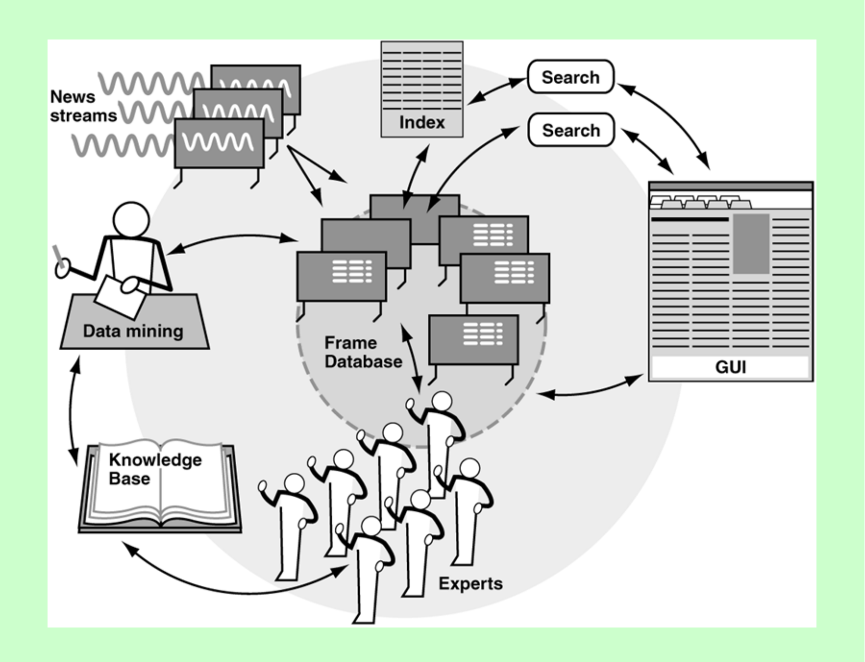

Table 1 Associations between Richard Butler and other proper nouns over a two week period in February. This is the complete rule list generated with a support requirement of 20 articles and a confidence requirement of 50%.

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

Table 1 Associations between Richard Butler and other proper nouns over a two week period in February. This is the complete rule list generated with a support requirement of 20 articles and a confidence requirement of 50%.

Richard Butler (28 examples)	Associations (% of co-occurrences)
Iraqi	(82%)
UNSCOM	(54%)

Table 2 Categorization results for South America

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

Table 3 Geographic rules proposed by data mining

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

View publication stats