

36

Self-Organizing News

by

Alan Wayne Blount

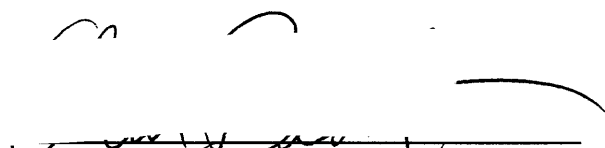
S.B., Computer Science and Engineering
Massachusetts Institute of Technology
(1991)

SUBMITTED TO THE PROGRAM IN
MEDIA ARTS AND SCIENCES,
SCHOOL OF ARCHITECTURE AND PLANNING
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 1993

© 1993 Massachusetts Institute of Technology

Author:

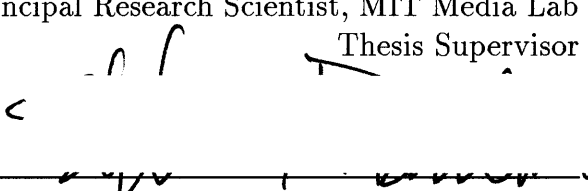


Program in Media Arts and Sciences
May 13 1993

Certified by:

Walter Bender
Principal Research Scientist, MIT Media Lab
Thesis Supervisor

Accepted by:



Stephen A. Benton
Chairperson, Departmental Committee on Graduate Students

Rotch
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

Self-Organizing News

by

Alan Wayne Blount

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on May 13, 1993 in partial fulfillment of the
requirements for the Degree of
Master of Science

ABSTRACT

Automatic newspaper design requires feature organization. An intermediary between an article database and the graphic designer is implemented that organizes features with regard to their style and content. Topics are determined and style assessed through simple keyword analysis augmented by various content-understanding heuristics. Articles are grouped by two means: comparison against predefined topics, and through classification by a clustering algorithm. The organizer maintains a history of the topics addressed and strives for continuity in its coverage of the news.

Thesis Advisor: Walter Bender
Principal Research Scientist, MIT Media Lab
This work was supported in part by IBM and the NIF Consortium.

Self-Organizing News

by

Alan Wayne Blount

Thesis readers



Reader:

Ken Haase
Assistant Professor of Media Technology
MIT Program in Media Arts and Sciences

Reader:

Victor McElheny
Director, Knight Science Journalism Fellowships

Contents

1	Introduction	6
1.1	Scope	8
1.1.1	Motivation	9
1.1.2	Approach	11
1.1.3	Expectations	16
2	Newspaper Specification	18
2.1	Sections	19
2.2	Retrieving Articles from Server	21
3	Experiments	23
3.1	Mixed Bag	26
3.1.1	Baseline	26
3.1.2	Improvements	30
3.1.3	Temporal Performance	32
3.2	Wall Street Journal	33

3.2.1	Keywords	34
3.2.2	Proper Nouns	35
3.2.3	Keywords and Proper Nouns	36
3.2.4	W.S.J. Analysis	36
3.3	Evaluation	37
4	The Demo System	39
4.1	Using bpaper	39
4.2	Implementation	41
5	Future Directions	44
6	Acknowledgments	45

Chapter 1

Introduction

Human newspaper and magazine wire-service editors select the features which they estimate best represent the interests of their readers and themselves. A newspaper that contains an article collection tailored to the interests of a specific reader, as well as material of general interest, may be more interesting than today's mass-market paper. Unfortunately, an editorial and design staff can lay out a single paper for sixty thousand people far cheaper than sixty thousand personalized papers. The cost of human editors and designers precludes their day-to-day use in personalized news. Can newspaper design be automated?

The automatic design of a newspaper may be split into three tasks. The first concerns news generation and selection. Today's traditional newspaper articles are written and placed into electronic form by the paper's staff or culled from the newswires by editors. A fully automatic newspaper lacks, of course, a reporting staff, but may

still present news from the newswires, as well as from other sources not available or frequently used by traditional papers. Non-traditional news items may include personal electronic mail, mailing lists, Usenet articles, weather and flight data, and on-line calendar entries.

Another task is page make-up. Articles must be placed upon the page in a pleasing manner, to catch the reader's attention and allow him to easily find his way through the day's news. A graphic designer's expertise is not easily quantified or mechanized, but systems have been devised that automatically lay out information, given a hierarchy of features. Colby's LIGA[11] is a prototype of such a system.

The third task fits logically between the content generator/selector and the page make-up system. Articles must be organized into the hierarchy that a system such as LIGA expects. More news is written daily than any person could read. The electronic newspaper must find the interesting articles and make them presentable. Print newspapers are sectioned—and within sections, the articles are organized with regard to priority and other relationships. The more important articles are placed at the beginning of a section, and given greater graphic weight. Related articles are juxtaposed. Topic threads that cover particular events in the news are carried day-to-day.

This thesis describes a system that organizes news articles with regard to their content and style, acting as an intermediary between the article database/user-modeling system and the graphic designer.

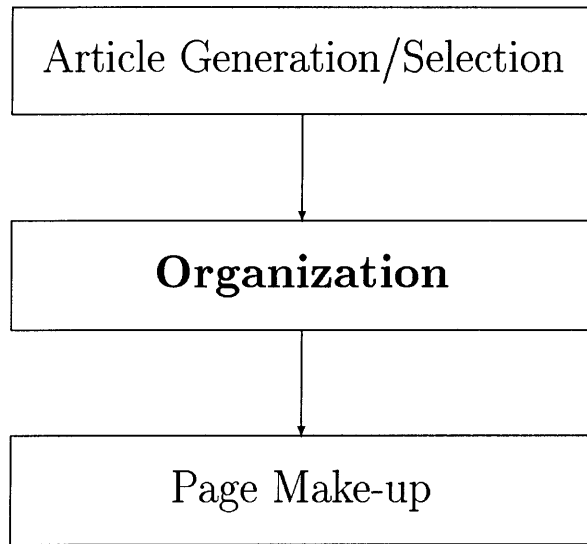


Figure 1.1: Newspaper design system.

1.1 Scope

The goal of this thesis is a system that automatically organizes electronic news. Such an organizer will be an integral part of future news distribution systems, whose structure and character may be very different from today's mass-market monolithic media. Bender *et al.*[7] describe the nature of this future news environment:

The newspaper industry... is unsurpassed in its ability to gather and organize vast quantities of time-sensitive information. The weakness of the industry lies in its outmoded distribution and presentation of that information, as well as its inability to be responsive to the needs of both individual readers and advertisers... The future of not only newspapers, but also all forms of information dissemination, lies in cooperation between the information provider and the audience, both of whom will be operating in a computationally-rich environment.

News articles may be sub-packaged at the source or at a news organization service, and finally amalgamated locally. The reader need not micromanage the day-to-day operation of her newspaper. The details will be handled by the her computer.

Such signals are not directed at a human recipient, but rather, to a local computational agent acting on her behalf. In response to instructions from *both* the distributor of the signal and the reader, this agent operates upon the signal in manners both suggestive of traditional media and of new forms. Digital electronic media should give us the opportunity for forms of data distribution that invites user participation with the data.[8]

The following sections describe the motivation for organized electronic news, a means of achieving it, and some particular expectations.

1.1.1 Motivation

The well-built newspaper takes on a life of its own in the mind of its readers. Berry[9] eloquently describes the association that may result between paper and reader:

The habitual reader soon comes to know his newspaper as thoroughly as he knows a close associate. He learns to recognize the kind of stories used, the stylistic practices employed, and the attitude of the paper on all the prominent questions of the day. He also discerns a pattern in the use of headlines, banner lines, punctuation, and typography; and he sees a story-to-story and a day-to-day similarity in the presentation of the news...Eventually this reader tends to think of the newspaper as a personality, as a collection of writings emanating from a single mind rather than from many minds; and he quotes his newspaper as if it were a person, approving or reproving it as he would a member of his family.

Today's electronic papers are far from what most readers would consider a member of their family. Lie's Electronic Broadsheet[15] is capable of presenting several complete articles at once on a large screen, but the news lacks any story-to-story or day-to-day continuity beyond simple sectioning based on information provided with each story. The paper places articles tagged "International" by the news supplier on

the International page, but does not design Thursday's paper with knowledge of the news it presented on Wednesday, or attempt to graphically associate related items.

Two general types of organization are necessary to create a readable electronic newspaper. Topical sectioning is one method. Traditional papers are sectioned—Nation/World news before Entertainment before Sports. Some ongoing columns and other features are always placed statically relative to other features: The Boston Globe's comics always appear next to "Ask the Globe" and the chess puzzle. Two interpretations of the same event are often placed near one another.

Articles may be grouped according to their style as well as their subject. Consider the following case from traditional news: Two articles address a football game. One is a sportswriter's analysis, the other a table of the players' statistics. Given that many readers may have no specific interest in the statistics, they are usually relegated to an "inside page" of the sports section, grouped with other articles containing heavy statistics, while the analysis may rate the top story.

Stylistic association is likely to be a more difficult goal for electronic papers than their traditional counterparts. If a newspaper editor decides an article requires a sidebar explaining the terms or statistics used in a story, the editor will assign the task to a staff writer who will create the piece. Automatically-assembled papers lack the luxury of a reporting staff. Further, today's electronic news feeds rarely contain prepackaged article clusters. Such packaging is an editorial, rather than reporting task, and must be handled by the paper design system.

Electronic papers may present a heterogeneous view of news. Consider a paper that mixes articles originating in The New York Times and USA Today. The difference in the reading level and depth of coverage may be jarring to the reader if the articles are mixed indiscriminately upon the page. Editors will often place a short, stylistically simple summary alongside a deeper treatment of an event. Articles posted to Usenet are self-edited and of wildly varying writing quality and information content. The Usenet world longs for a “flame detector” to flag articles that contain undue amounts of idiosyncratic opinion.

Though the page layout system is not the subject of this thesis, its character is of some interest to the organizing agent. The electronic paper may be formatted like traditional printed newspapers, with a dense front page and sectioned into general topics. On-screen papers, however, do not require “inside pages.” Dienes suggests a paper that has *only* a front page and reformats itself with the most relevant remaining features each time a feature is finished, reasoning that “You only have one page on a screen—you might as well make it the front one.”

1.1.2 Approach

The relationships used in organizing a paper can only be determined through knowledge of content. A human editor can readily determine whether two articles address the same topic. Automatic article understanding is an open-ended problem in computer science. There are several approaches, the simplest of which is keyword recog-

dition. An article that contains the word “Bush” may have to have something to do with the former President. Or shrubbery. Most database systems allow the searcher to specify multiple words, such as “President, George, Bush,” to aid in differentiation. Bettyserver, the news article database used for this thesis, allows quick searches for articles that contain certain keywords, as well as boolean searches on keywords. For example, all articles that contain the words “teflon” and “cabbage,” or the articles that contain “teflon” or “cabbage” may be retrieved with a single request.

The promises of natural language understanding beyond weighted keyword processing have remained largely undelivered. Methods of article understanding at a conceptual level have been implemented, but they are limited in scope. Alvarado[3], for instance, designed a large system that attempted to understand editorial text, but the system was limited to answering articles regarding U.S. protectionism. Salton[18] summarizes the problem:

In situations where the available texts span a variety of different topic areas, the conceptually-based text analysis methods that are often advocated in the literature are not usable, because it is impossible to build the knowledge bases that specify the contents and structure of the subject areas of interest.

Fortunately, keyword recognition and other techniques that use the texts themselves as the basis for analysis have been used with great success. Also, a great deal of ancillary information is supplied with many news sources. Most newspaper articles provided by the Clarinet UPI wire and USA Today come with the following:

- **category**
- **author**
- **publisher**
- **keywords** Supplied by the author or editor of an article.
- **date of publication**

Additionally, simple statistics and various heuristics may be applied to the body of an article to determine other relevant facts.

- **article length**
- **numeric vs. text content**
- **readability** Devereaux index uses word and sentence lengths to estimate readability.
- **proper name usage** May be estimated through a count of capitalized words.
- **keywords** Words that are indicative of the contents of an article may be guessed through frequency analysis and noiseword elimination.
- **breadth of interest** An article appearing in the USA Today feed is assumed to be of more general interest than an obscure Usenet “alt.” newsgroup.

Simple heuristics may be able to uncover information more subtle than one would expect. A recent Boston Globe Ombudsman’s report, for example, stated that over a period of several months the Globe used the term “right wing” when speaking about Republicans more often than it used the term “left wing” when speaking of Democrats. An article organizer can make use of statistics such as these when estimating the political slant of a feature writer.

Assume we have made judgments about the articles to be organized and have some content indices. For example, the numeric content, reading level, and political slant (liberal vs. conservative) are expressed as integers. Assume also that we have isolated critical words or phrases from the text that have some bearing on its content (noiseword elimination). Two approaches to forming an article organization from this information are evident:

1. **Predetermined Topics** An obvious attack is to compare the article statistics and keywords against statistics and keywords known to be associated with a given topic. For instance, articles on foreign aid may be defined as all articles from sources “clari.news.international” or “usa-today.international” containing the words “foreign” and “aid.” All articles that satisfy the conditions of the definition are said to address foreign aid.

This system may produce arbitrarily robust clusters. Most articles come with enough ancillary editorial information to place them in coarse sections. All of the articles in source “clari.news.international,” for example, are suited for placement in the International section of a paper. This “Predetermined Topics” approach, however, does have a severe disadvantage. It is impossible to anticipate all the topics that may be covered in the news. The international news for a week, for example, may center on war in Bosnia and elections in Russia. Predetermined topic clustering does not work at a fine enough detail level to separate these topics, unless a human operator is watching the news

and updating the topic definitions as new events develop.

This thesis uses a predetermined topics system to group articles in sections. A set of topics are defined using source names, ranges of content index values, and typical keywords. The user or his user-modeling system chooses desired topics from this list. Articles *within* these sections are further clustered using an organizing system that requires no *a priori* knowledge of article topics, yet provides finer-grained topic clustering. It is described below.

2. **Unknown Topics** Assume we have determined several indicators of article content, such as the numeric content and political slant indices described above, as well as the proper nouns in an article. We may feed these data to a clustering algorithm.

A primary question regards which indices are best suited to article clustering. Articles clustered using a political slant metric may exhibit a more interesting coherence than those clustered by length. A group of articles selected for their radical ravings will likely be more interesting than a random group of long articles. Weights must be determined for each metric that reflect their relative clustering importance.

The indices that are best suited to clustering are those that exhibit a non-monotonic spread of data-points. Again article length is likely a bad choice. A graph of the lengths of articles against their number describes (approximately) a descending logarithmic curve, with few natural breakpoints. The curve for

numeric content, however, is likely to show peaks. Ordinary news articles and stories have few numbers, while sports scores and financial charts have many. Clustering on numeric content will likely enable identification of running features containing large numeric content.

Since news reception is sequential, the article organizer must maintain a history of the topics that have recently appeared in each paper generated and attempt to keep the reader updated on these topics as new information arrives (unless the reader indicates his dislike for a thread). This organization over time is conceptually similar to organization over space, different only in the implementation details. The greatest challenge is to recognize new topic threads and track the progress of ongoing topics.

Once articles are clustered, they are passed to the graphic designer.

1.1.3 Expectations

The goal of this thesis was an electronic newspaper demonstration system that creates a paper that is pleasant to read. I hope that, in addition to being used in demonstrations, it will be used for casual news reading.

To sum up, the organized electronic paper has the following features:

- Intelligent grouping of articles into sections using predetermined topic definitions and within sections by unsupervised clustering.
- Edition-to-edition continuation of topic threads.

Essential information to be determined through experimentation includes the ideal relative weights of various clustering criteria.

Chapter 2

Newspaper Specification

To aid understanding of the workings of the newspaper generator *bpaper*, it is expedient to first describe *bpaper*'s output. The following chapter describes a specification for an organized paper.

Some definitions must be given. The implementation of this thesis, *bpaper*, makes use of Abramson's `dtype++` library[2]. This library provides a list data class, with functions to create, store, retrieve, and exchange lists of numeric and textual data. *Bpaper* stores its state and produces output in the form of `dtype` *environments*, which are lists of *bindings*. Each binding is a list of two elements, a *keyword* and a *value*. `("name" "Blount's Paper")`, for example, is a binding that maps the keyword string "name" to the value string "Blount's Paper." The value may be of any data type, even another environment. Figure 2 contains a representation of a newspaper as created by *bpaper*.

Bpaper takes its news from *bettyserver*[10], an augmented news article server. News articles are also stored in dtype environments. The article format is described in [12]. Note that the papers created by bpaper have both `type` and `source` fields, and thus are valid news articles and may be submitted to a *bettyserver* for storage and later retrieval.

The following keywords are specified to have the following meanings within a bpaper:

- **type** The type of a bpaper is specified as ("`item`" "`newspaper`" "`bpaper`"), with "newspaper" a subclass of "item" and "bpaper" a subclass of newspaper.
- **name** The name of the paper, to be printed at the masthead.
- **username** The owner of the paper.
- **email** The owner's electronic mail address.
- **host** The host running the news article database from which this paper was created.
- **port** The port number of *bettyserver* on **host**.
- **source** Always given as "bpaper."
- **date** The paper's creation date, in human-readable format, to be printed on the masthead.
- **sections** The body of the newspaper. Described in greater detail below.

2.1 Sections

Newspapers built by bpaper have two levels of article organization. Articles are first organized into sections, much like sectioning in traditional newspapers. Within each

```

(("type" ("item" "newspaper" "bpaper" ) )
 ("name" "Blount's Paper" )
 ("username" "blount" )
 ("email" "blount@amt.mit.edu" )
 ("host" "monk.media.mit.edu" )
 ("port" 11119 )
 ("source" "bpaper" )
 ("date" "Wed Apr 21 20:56:59 1993" )
 ("sections"
  (((("name" "Russia Today" )
     ("threads"
      (((("thread_id" 2 )
         ("arts"
          ("

```

Figure 2.1: Sample newspaper specification with two topics

section articles are organized into threads. Each thread nominally addresses a single separate topic within the day's news.

The body of a `bpaper` is described within a nested environment within the `sections` binding. Each section is a list of two bindings. They are the section's `name`, and a nested environment containing the `threads` within that section. Each thread has an integer thread identifier, or `thread_id`, and an `arts` binding that contains a list of the uniqueids of each article within a thread. Note that the articles themselves are not placed within the paper specifications, instead merely their uniqueids, in order to save storage space and eliminate redundancy.

`thread_ids` are unique identifiers within sections, and are guaranteed to uniquely identify threads over time. A newspaper layout agent, for example, may place a section's articles from `thread_id` 7 in the upper-left corner of a page. `Bpaper` will group future articles in that section believed to be associated with that topic in thread 7. The layout agent is free to do what it will with that information, such as place the new issue's articles on that thread in the same general location, or use some other graphic element to visually group the articles.

2.2 Retrieving Articles from Server

The articles identifiers are given with long-form unique ids. A layout engine may retrieve the actual articles from the server at the given `host` and `port` with this sequence of requests:

(lookup uniqueid [uniqueid]) //returns *short-form uid*

(getart uid) //returns *article*

The first call will return a DT_NULL if the article has expired.

Chapter 3

Experiments

Initial work in article clustering was accomplished by Salton as early as 1968[17]. Salton's SMART Retrieval System was a first attempt at using computers to search through large numbers of documents for those that match a given request. The subject matter was varied; documents searched were from the fields of computer science, documentation, aerodynamics, and medicine:

The system takes documents and search requests in English, performs a fully automatic content analysis of the texts, matches analyzed documents with analyzed search requests, and retrieves those stored items believed to be most similar to the queries.

Article clustering was done not to enhance presentation but instead to improve search efficiency. Clustering involved initially comparing a number of criteria in each article against every other article and grouping those articles which are sufficiently similar. Each cluster would be represented by a *centroid vector*[17], which contains keyword frequencies for the articles within a cluster. In a database of several thousand

articles, one would compare a search request first against the centroid of each cluster. Articles in clusters with sufficiently similar centroids were then compared against the request. This technique results in a savings of processing time equal to the average number of articles per cluster.

Similar techniques may be used to cluster news articles for presentation in an electronic newspaper. Each article to be placed in a paper may be compared against the centroids of the clusters already in place. The article is placed in the cluster that best matches the article's characteristics. Note, however, that the ends are not identical: In Salton's system, articles could be readily placed in more than one cluster. In a newspaper, assuming that readers only want to read each story once, articles need only be placed in one cluster.

As mentioned in Chapter 2, *bpaper* utilizes a two-tier approach to article organization. In the first pass, *bpaper* retrieves articles from the server that are appropriate to a given section. All of these predefined topics are maintained by the server in a list. When the prototype of a newspaper is created, the reader or the reader's user-modeling system specifies the topics that the reader is interested in. These topics may correspond to a particular news source, such as "clari.news.top," or to articles matching a boolean keyword query, or some combination of the above.

Bpaper retrieves all the articles currently available for each topic. If more articles are retrieved than a maximum-per-section specified by each paper, extra articles are discarded, in no particular order (*bpaper* makes no judgments about article suitability).

ity). These articles may be picked up by the next issue, if they have not expired.

The articles for a section are then clustered. The experiments described here cluster articles using a simplified form of Salton's keyword algorithm coupled with additional heuristics. Salton's system uses a normalized term weighting function that assigns a weight to each term proportional to its frequency in a document times the inverse of its frequency in the document collection, divided by a length-normalizing factor.

Salton's document collection was relatively fixed. The overall frequency of each term in the document collection could be determined once and stored. In a news retrieval system, where the database is updated hourly and turned over weekly, term frequencies would need to be recomputed with each incoming set of articles, which is computationally expensive. Thus a simpler approach was opted for. The experiments described here cluster on keywords using only the frequency of terms in a document, their number normalized for document length.

The clustering algorithm works as follows: Each article is compared against the centroids of all the clusters in place by using several tests, including keyword matching. The centroids contain keywords, headlines, numeric content, and other data. Each of the results of these comparisons is multiplied by a weight, and the total is summed. If the total is greater than a threshold, the article is placed with the cluster that has the highest score. If not, the article starts a new thread.

The following sections describes the process in detail, and the results when applied to two newspapers: a mixture of UPI and Usenet articles, and an issue of the Wall Street Journal.

3.1 Mixed Bag

The tests were run on a mixed bag of four news sources with widely varying textual characteristics: Usenet personal advertisements, UPI commodities, disaster news, and baseball stories and statistics. The articles were drawn from the news supplied to Bettyserver on April 9 and 10, 1993. The results of all the tests are presented in figures 3.1.1, 3.1.1, 3.1.1, and 3.1.1.

3.1.1 Baseline

The baseline experiment was run clustering on keywords only, with all other test results assigned a weight of zero. Each article is augmented before it is supplied to the news server with a list of the fifteen most frequent words found in the body of the text after *noisewords*, words deemed to have little bearing on the content of an article, are eliminated. The noisewords in the filter are found in Figure 3.1.1.

The first task was tuning the keywords match threshold to a number that would result in a balance between appropriate stories being rejected and inappropriate stories being accepted. Four, the number used by Bender and Chesnais in Network Plus[6], was found to give acceptable results after noiseword rejection.

a able about after all along already also although am an and another any are arent as at b bad be because begin been before behind below beneath between by billion but by c can cant char come could d day did do does doesnt doing dont down dr due during e each easy eight else end every everyone f few fill first five for four from front g get go goes good going great h had hard has have he here hereby hers his how hundred i if in int into is isnt it its j just k know l last let lot m main make many may me might million miss more mr mrs much must my n new next nine no none not now o of off on once one only or other our out over p people percent place procedures put q r return s said same say second seven she shell should since six so some such sure t tell ten than that thats the their them then there therefore these they theyre thing third thousand this those though three through to too top trillion two under u until up upon us use used v very w was we week well went were what when whenever where wherever which whichever while who why will with without wont would x y year yes yet you your z zero

Figure 3.1: Noisewords list

Column 1, labeled “K,” gives the results of clustering using keywords only. This may be compared against the column labeled “ED,” which lists the clusters in which the author believes articles should be placed. Note that in Figure 3.1.1, Disaster News, keyword clustering produced optimal results. In Baseball News, keywords grouping also produced some interesting clusters, with three articles on the White Sox and White Sox player Bo Jackson grouped together, and four articles on the Dodgers clustered as well. The Personals did not fare so well, with 15 topics identified, four more than the target of 11.

Standardized article headers and trailers present a problem for keyword-based clustering. Many Usenet article posters make use of one of several anonymous posting services. These services place a large advertisement for the service at the end of the article, which introduces false hits. These ads can of course be identified and eliminated from the keyword search system, but special cases such as this require maintenance as services come and go.

Test Case: alt.personals				
K	KH	KHN	ED	Headline
1	1	1	1	PC
2	2	2	2	Hot Lesbian Apologizes
2	2	2	2	Hot Lesbian Apologizes
3	3	3	3	Poor White Knight
4	4	4	4	What Rai would do in Reverse
5	5	5	5	Who are half you people
6	6	4	6	you're all sick
6	2	2	2	Hot Lesbian Apologizes
7	7	6	7	Phrases that should die FIRST
8	8	7	8	SWF looking for SWM friends
9	9	8	7	Phrases that should die
9	9	8	7	PMS every day??
10	10	9	8	10,000 Maniacs
10	10	9	7	Where's Jessica
11	9	8	7	PMS every day??
12	11	10	8	Chocolate
13	12	11	9	SWM ISO S/DWF in Chicago
14	13	12	10	More Pointless Animals
15	14	13	11	What the F**K

Figure 3.2: The Personals. Key: K: keywords only; KH: Keywords and headline; KHN: Keywords, headline, and numeric; ED: Author's pick.

Test Case: clari.biz.commodity				
K	KH	KHN	ED	Headline
1	1	1	1	Farming Today
2	2	1	2	SLUG: packer-hogs
2	2	1	2	Georgia Cattle Prices
2	2	2	2	<Georgia-poultry>
2	2	1	3	For the week ending April 9 1993
3	3	3	4	American Metal Market Prices
4	1	1	1	Farming Today
4	1	1	1	Farming Today

Figure 3.3: Commodities

Test Case: clari.news.disaster				
K	KH	KHN	ED	Headline
1	1	1	1	Drowned teens returned home to NYC
1	1	1	1	Funeral for drowned brothers planned for after Easter
2	2	2	2	Cult claims six members dead
2	2	2	2	Koresh sends "Letter from God" to FBI
3	3	3	3	Water purification change may have caused contamination
3	3	3	3	Milwaukee officials encouraged by water tests
3	3	3	3	Great Chicago Flood, one year later

Figure 3.4: Disaster News

Test Case: clari.sports.baseball				
K	KH	KHN	ED	Headline
1	1	1	1	Indians 15, Yankees 5
1	1	1	1	Red Sox 9, Royals 4
1	1	1	2	American League Roundup
1	1	1	1	Yankees 11, White Sox 6
1	1	1	1	Cubs 7, Phillies 7
1	1	1	1	Rockies 11, Expos 4
1	1	1	1	Blue Jays 13, Indians 10
2	2	2	1	Braves 6, Dodgers 1
2	2	2	2	National League Roundup
2	2	2	4	Dodger bar is here to stay in Brooklyn
2	2	2	4	Dodger's Worrell placed on DL
2	2	2	1	Braves 2, Dodgers 0
3	3	1	3	Major League Boxscores
3	3	3	5	Saturday Probable Pitchers
4	4	1	3	National League Boxscores
4	4	1	3	American League Boxscores
5	5	4	6	Jackson homers in first at-bat
5	5	4	6	White Sox' Raines goes on DL
5	5	4	1	Yankees 11, White Sox 6

Figure 3.5: Baseball

3.1.2 Improvements

Two additional clustering techniques were tried as improvements over using keywords alone. The articles' headlines and their percentage numeric content are compared. The result of each comparison is multiplied by a weight before summation and comparison against a threshold, so that the importance of each test may be varied against the others to produce the best overall clustering. The following sections describe the results.

Headlines

Headlines are an obvious indication of content, and are particularly useful in clustering Usenet articles. In Usenet, articles which are replies or follow-ups to other articles generally have identical headlines¹. A range of weights was tried for articles with headlines that match a headline within a cluster. A very high weight results in articles with matching headlines always being placed in the same cluster, while a lower weight gives similar keywords greater effect. A matching-headline weight of five, equivalent to five matching keywords, keeping the minimum-match threshold at four, was found to give good clustering. Note that an article with six keywords in common with cluster X and a headline but no keywords matching cluster Y will be grouped with X over Y.

The results are presented in the the columns labeled "KH." The Personals and

¹Of course, in Usenet articles threads are usually given explicitly in the "References:" field, though this is not always the case.

Commodities sections showed improvement.

Numeric Content

Percentage numeric content is the percentage of an article's characters that are numerals, rather than punctuation or letters. It is a particularly useful metric for presentation clustering. Articles containing mostly numeric data, such as stock quote listings, are typically given their own section in traditional papers.

The articles in the group "clari.news.baseball" exhibit numeric content statistics as given in figure 3.1.2. Notice that there are two main clusters, those articles with numeric content centering around 1% and those with content around 10%. This is due to the nature of sportswriting: Articles are usually either text treatments of events or lists of statistics. The same is true of financial writing. A breakpoint of 8% was chosen as a good point between text-heavy and number-heavy articles.

The column labeled "KHN" lists the clusters that result when a weight of four is given to articles with greater than 8% numeric content, retaining weights of five for identical headlines and one for each matching keyword, with a threshold of four.

Several improvements in clustering are evident. In Commodities, Farming Today, packer-hogs, and Georgia Cattle prices are grouped together, all articles with large columns of data. Baseball also shows improvement, with the Boxscores grouped together, though several game report articles are placed in with the Boxscores. Personals and Disaster News were unaffected.

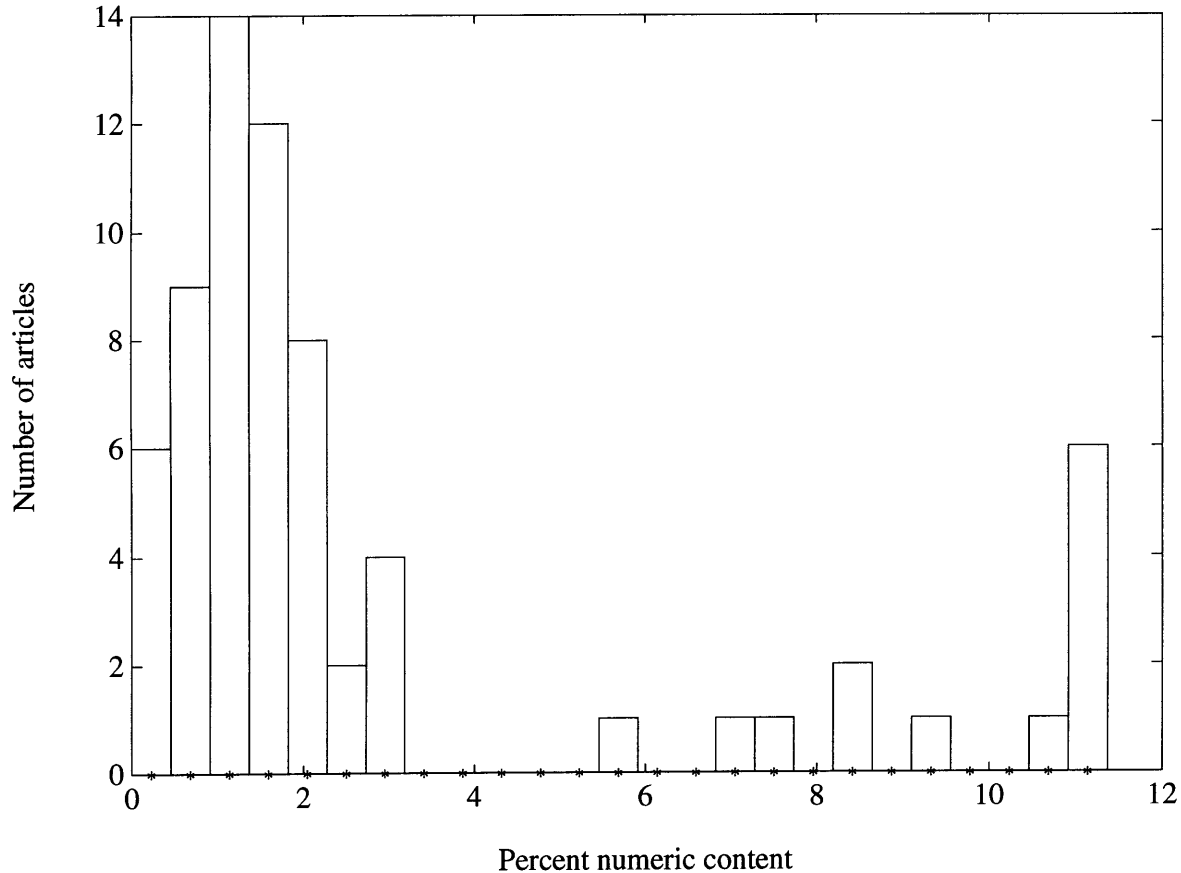


Figure 3.6: clari.news.baseball percentage numeric content histogram

3.1.3 Temporal Performance

Two further tests were run, using the cluster database generated by the full clustering described in Section 3.1.2 above and listed in the charts under “KHN.” The first test clustered articles received immediately after those listed in the figures, and the second articles received two weeks later. The first set clustered 39 articles, with 15 of the articles falling into eight new clusters overall. The two weeks test produced considerably more clusters, with 56% of the articles falling into new clusters. This is to be expected, however, as news changes with the times.

The system maintains no upper limit on the number of clusters that may be created. Processing time will tend to grow with each day’s news unless inactive

clusters are removed. Clusters that have not grown within a week or two are likely safe to be forgotten, as the reader will probably not remember the cluster's placement in the paper anyway.

3.2 Wall Street Journal

A second suite of tests was run, in an attempt to determine how automated clustering compared to a professional editor's organization. Electronic and printed versions of the May 4, 1993 Wall Street Journal were obtained. Unfortunately, article overlap between the two editions was not complete. The printed version had 69 of the 103 articles that were in the electronic version, and the electronic version was missing the stock listings as well as a number of articles found in the printed. Regardless, there was enough overlap to make meaningful comparisons of the clustering.

The strongest clustering in the printed Journal were the articles in their Business Brief and Who's News sections, which link are collections of short articles. The electronic feed does give some explicit sectioning information. Headlines of each article are usually prepended with a section title if the article is to be placed in a running section: Business Briefs, for example, have headlines of the form "Business Brief – *Headline*." For the purposes of these tests, however, this information was ignored.

Tests used keyword clustering, proper noun clustering, and a combination of the two. Headlines and numeric content contributed nothing to these tests, as there

were no identical headlines and Journal features with large numeric content were not present in the electronic version.

Several clusters were found in the printed version, and were used as controls for comparison of cluster quality in the various clusterings described below. They are as follows:

- **Health Care** Though not grouped in an explicit section, three articles on health care were present: Two on Clinton's health plan and a third on Germany's attempt to contain health costs. The first article was linked to the other two with an inset box.
- **Who's News** Short articles on business personalities.
- **Business Brief** Short articles on business news.
- **Leisure and Arts** A book review, a music review, and a set of capsule movie reviews.
- **Labor** An article on a mine union's dispute with BCOA is placed next to an article on Northwest air winning labor concessions.

3.2.1 Keywords

The first attempt at clustering was run using keywords only. The keyword threshold was varied from three to five. The total number of resulting clusters is given in figure 3.2.1.

- **Health Care** With a threshold of three words, two of the three health care articles share a cluster. The health-care focusing on the White House is grouped with an article on the Clinton Staff and an editorial defining democratic moderates. No effective clusters with 4 or 5 word thresholds.

Threshold <i>vs.</i> Number of clusters		
Threshold	keywords	proper nouns
3	61	58
4	80	68
5	88	78

Figure 3.7: Threshold *vs.* No. of clusters

- **Who’s News** A threshold of 5 split these 11 articles into 6 groups, while a threshold of 3 placed them in 3 groups.
- **Business Brief** A threshold of 5 split 16 articles into 11 groups, while a threshold of 3 placed them in 5 groups. The Journal itself used three of these articles as space fillers at the bottom of pages, outside of the Business Briefs section.
- **Leisure and Arts** No threshold produced a cluster containing more than one of these articles.
- **Labor** Articles are clustered when the threshold is set to 3.

3.2.2 Proper Nouns

Each capitalized word that is not at the start of a sentence is designated a proper noun. This technique is not wholly reliable, but is algorithmically simple and gives good results. False positives account for only about 10% of the proper nouns determined by this technique performed on the May 4 article set. They were clustered exactly as keywords, with the following results:

- **Health Care** No effective clusters.
- **Who’s News** No articles were clustered together, even with the threshold set as low as 3.
- **Business Brief** A threshold of 3 produced 10 clusters.

- **Leisure and Arts** No threshold produced a cluster containing more than one of these articles. The book review, *Images of East Germany*, was placed with two other articles on Germany, however.
- **Labor** Articles are not clustered.

3.2.3 Keywords and Proper Nouns

A combination of the above techniques was tried, giving each matching keyword or proper noun an equal weight of 1 with a threshold of 4.

- **Health Care** Two of the three articles share a cluster.
- **Who's News** 11 articles split into 4 groups.
- **Business Brief** 16 articles in 9 groups, clustered with 7 additional non-Business Brief articles.
- **Leisure and Arts** Book review placed with one other article on Germany.
- **Labor** Articles are placed in same cluster.

3.2.4 W.S.J. Analysis

The Who's News and Business Brief results are especially encouraging. It is not surprising that keyword clustering is particularly successful with the Who's News articles, as they all follow a similar form, describing executives being named to various positions.

The generally lackluster performance of proper noun clustering in the above tests is not wholly telling. The printed Journal clustered a review of the book *Images of East Germany*, a set of capsule movie reviews, and a music review together on the

Leisure and Arts page. In none of the above tests were the three articles clustered together, as they had no textual content in common. Some external knowledge base strategy would likely be required to properly recognize the three articles as belonging to Leisure and Arts. The book review, however, was placed with two other articles on Germany.

The combination of methods retained many of the best features of the two, but at the cost of many unrelated articles appearing in the targeted clusters. It is noteworthy that the Labor articles were successfully clustered with a threshold of 4. Keywords-only clustering failed with a threshold of 4, which indicates that a proper noun contributed to the cluster hit.

3.3 Evaluation

Hard criteria for the success or failure of an organization system are difficult to determine. Given two dozen news items, there may not exist a single best organization that two human editors would agree on. Fortunately, bad organization is readily apparent. An article on George Bush in the “Home and Garden” section of the newspaper is an obvious failure, funny the first time but eventually tiresome.

In the proposal for this thesis, I specified that I would consider an 80% success rate acceptable for an organizer, as determined by comparing articles categorized and organized by a human editor against the algorithm’s organization. In the “mixed bag” test using keywords, headlines, and numeric data, 41 of 53 or 77% of the articles

were placed in the desired clusters.

Ideal clustering depends entirely on what the individual reader wants to see. A baseball fan interested in an overview of the games of the day would probably like to see all of the “Indians 15, Yankees 5” style game reports grouped together, while a reader interested in a specific team may want the stories on that teams individual players grouped with the game reports. The paper does exhibit continuity from day to day.

Of course, this work in relating articles is only the beginnings of a newspaper that will compete with human-organized editions. In many cases, an human editor would want related articles placed *away* from one another, or dropped entirely from an issue, to eliminate repetition. Herbert Gans[14] lists five criteria for design *balance*, used in choosing and organizing features:

- Story Mixture
- Subject Balance
- Geographic Balance
- Demographic Balance
- Political Balance

With a readership of one, *balance* may be replaced by the *accuracy* with which features are chosen and these criteria will still hold. While this thesis is only directly concerned with story mixture, the other four criteria are at least as important in producing a compelling paper.

Chapter 4

The Demo System

Bpaper, the newspaper server, was designed to be both a research tool and a low-maintenance demo engine.

4.1 Using bpaper

Bpaper has three optional arguments. ‘-h’ and ‘-p’ are used to specify the hostname and port of the bettyserver where bpaper will get its news. ‘-d’ is used to specify the top level directory where bpaper is to store its state files.

When bpaper is run it immediately attempts to connect to its bettyserver, and if it is successful, sits and waits for clients to connect. Figure 4.1 lists the commands that clients may issue to bpaper.

A paper specification for a new user may be created by issuing the `subscribe` command. *name* and *username* are strings, and *topics* is a list of keywords that

(generate <i>username</i>)	Generates a paper for user
(upload <i>username</i>)	Uploads paper to bettyserver
(reset <i>username</i>)	Resets paper history for user
(subscribe <i>name username topics</i>)	Creates new paper specification
(help)	Print help message

Figure 4.1: bpaper requests

define the topic sections the reader is interested in.

Dienes' layout system within Demo Box, the Garden demonstration system, calls the server and issues a generate command for a requested newspaper. It displays articles under section headings with articles of the same thread printed on the same color background. It is hoped that this demonstration system will be improved to place related articles near one another and use lines, headers, and other graphic elements in grouping articles in threads.

A paper may also be read with bread (pronounced b-read) on any ASCII terminal. Bread produces an ASCII paper from a bpaper specification. Bread does nothing clever with the section or thread information, but merely identifies each article with its thread id. Bread has two options: '-f' forces bread to print each article in its entirety (the default is only the first 512 characters). '-h' forces bread to print only the header information from each article.

4.2 Implementation

Bpaper maintains its state in two files. The topics file contains a list of all the predefined topics that the server is aware of. These topics are stored in a single large dtype environment. Each binding maps an integer topic id to a nested environment.

The following three keywords are defined for use in these nested environments:

- **name** The name of a topic, properly capitalized and suitable for placement at the head of a section in a newspaper.
- **keys** Words that describe the gist of a topic. For example, appropriate keywords for a topic on government corruption may be “government,” “corrupt,” and so on.
- **bettyreq** The request that is sent to bettyserver to retrieve articles on this topic.

The current implementation of bpaper uses a list of about 200 topics from which a user may choose.

The second file used by bpaper is the papers file, which contains another environment. The following keywords are used:

- **host** The host running the news article database where bpaper gets its news.
- **port** The port number of bettyserver on **host**.
- **weights** Weighting assigned to each clustering method, as described in Chapter 3.
- **threshold** Minimum number of points required for an article to match a cluster. Described in Chapter 3.
- **papers** Value is a list of environments, each describing the state of a paper. These environments are further defined below.

```

(((name "Blount's Paper")
  (username "blount")
  (email "blount@amt.mit.edu")
  (max_articles_per_section 10)
  (sections ((2 ((delivered (uid1 uid2 uid3 uid4 uid7 uid8 uid10))
    (max_id 2)
    (threads ((1 ((keywords (foo bar))
      (headlines (head1 head2))))
      (2 ((keywords (baz quux))
        (headlines (head1 head2))))))))))
  (202 ((delivered (uid1 uid2 uid3 uid4))
    (max_id 1)
    (threads ((1 ((keywords (foo bar))
      (headlines (head1 head2))))))))
  (6 ((delivered (uid7 uid7))
    (max_id 0)
    (threads ((1 ((headlines (head7 head8))))))))))

```

Figure 4.2: A paper as given in the papers environment

Bpaper is implemented as a hierarchy of C++ classes that mirror the structure of a newspaper. The top level is the *papers* class, which contains data and procedures relevant to all the newspapers in the server. Below this class are the *paper*, *section*, and *thread* classes which create and manage the various elements of each paper. The code is commented and compartmentalized, and may be maintained with little effort.

Chapter 5

Future Directions

Much may be done to improve article clustering. Keyword clustering may be improved by implementing Salton-style term weights, though at a heavy computational cost. *Word-stemming*, eliminating suffixes of keywords to eliminate redundancy, will also increase keyword clustering performance[17].

More drastic improvements in article clustering will result from using better indices of content. The political slant index mentioned in the introduction would be a worthy project, certainly of use in clustering.

Thread management over time needs improvement. Inactive threads should be eliminated to improve clustering efficiency.

User feedback on the may be valuable. If a user may readily inform bpaper an inappropriate article, that article's influence can be removed from the cluster centroid so that future articles will be better clustered.

Chapter 6

Acknowledgments

The author would like to thank the following people: My parents, Joyce and Carey Blount, and my brother Andrew. You have my love and gratitude for all the support you've given me over the years. Walter Bender, my advisor, is a bottomless pool of insight, and makes sure people have some fun around the Garden. Thanks to my readers Professor Haase and Victor McElheny for taking the time and effort. Thanks Lacsap for hiring me way back in '88, and Linda for putting up with me.

Thanks to Nathan, Orwant, Klee, and the rest of the garden Netrek crew, as well as the authors of Netrek and the server gods who keep it running. Finally, thanks to all current and former members of BDA and Psychic Hotline. What a band.

Bibliography

- [1] Abramson, Nathan. Context-Sensitive Multimedia. Master's thesis, Massachusetts Institute of Technology, 1992.

- [2] Abramson, Nathan. dtype++ — Unification of commonly used data types. Technical report, Electronic Publishing Group, MIT Media Laboratory, October 1992.

- [3] Alvarado, Sergio J. *Understanding Editorial Text: A Computer Model of Argument Comprehension*. Kluwer Academic Publishers, 1990. *Conceptually-based natural language system applied to editorial text*.

- [4] Andrew Jennings, Huan Liu, Hideyuki Higuchi. A Personal News Service Based on a User Model Neural Network. In Judy Kay, Alex Quilici, editor, *Proceedings of the IJCAI Workshop W.4 Agent Modelling for Intelligent Interaction*, August 1991. *Augmented keyword search on news articles*.

- [5] Ball, G. H. and Hall, D. J. Isodata, A Novel Method of Data Analysis and Pattern Classification. Technical report, Stanford Research Institute, Stanford,

- CA, 1965. *Describes effective clustering algorithm.*
- [6] Bender, Walter and Chesnais, Pascal. Network Plus. In *Proceedings, SPIE Electronic Imaging Devices and Systems Symposium*, pages 81–86, January 1988.
- [7] Bender, Walter, Cooper, Muriel, Davenport, Glorianna, Haase, Ken, and Negroponte, Nicholas. The Newspaper of the Future: A Straw-Man Proposal in Four Parts. Technical report, MIT Media Laboratory, July 1991.
- [8] Bender, Walter, Lie, Hakon, Orwant, Jonathan, Teodosio, Laura, and Abramson, Nathan. Newspace: Mass Media and Personal Computing. In *USENIX Conference Proceedings*, Nashville, TN, June 1991.
- [9] Berry, Thomas Elliott. *Journalism in America*. Hastings House, 1976. *Covers newspaper organization and design.*
- [10] Blount, Alan. Bettyserver: More news than you can beat with a stick. Technical report, Electronic Publishing Group, MIT Media Laboratory, December 1991.
- [11] Colby, Grace. Intelligent Layout for Information Display: An Approach using Constraints and Case-based Reasoning. Master’s thesis, Massachusetts Institute of Technology, 1992. *A method to create graphic designs for structured information.*
- [12] Dienes, Klee. Bettykit. Technical report, Electronic Publishing Group, MIT Media Laboratory, December 1991.

- [13] Evans, Harold. *Newsman's English*. Holt, Rinehart, and Winston, 1972. *The structure of a news story*.
- [14] Gans, Herbert J. *Deciding what's news*. Vintage Books, 1979.
- [15] Lie, Hakon Wium. The Electronic Broadsheet - all the news that fits the display. Master's thesis, Massachusetts Institute of Technology, 1991. *Describes a personalized electronic newspaper*.
- [16] Orwant, Jon. Doppelganger. Master's thesis, Massachusetts Institute of Technology, 1991. *Personalized news system is described*.
- [17] Salton, Gerard, editor. *The SMART Retrieval System, Experiments in Automatic Document Processing*. Prentice-Hall, Inc., 1971.
- [18] Salton, Gerard, Buckley, Chris, and Allan, James. Automatic Structuring of Text Files. Technical Report TR 91-1241, Department of Computer Science, Cornell University, Ithaca, NY, October 1991.
- [19] Therrien, Charles W. *Decision Estimation and Classification*. John Wiley and Sons, 1989.