# Classifying News Stories using Memory Based Reasoning

**Brij Masand, Gordon Linoff, David Waltz***

**Thinking Machines Corporation**
**245 First Street, Cambridge, Massachusetts, 02142 USA**

## 1 Abstract

We describe a method for classifying news stories using Memory Based Reasoning (MBR) (a *k*-nearest neighbor method), that does not require manual topic definitions. Using an already coded training database of about 50,000 stories from the Dow Jones Press Release News Wire, and SEEKER [Stanfill] (a text retrieval system that supports relevance feedback) as the underlying match engine, codes are assigned to new, unseen stories with a recall of about 80% and precision of about 70%. There are about 350 different codes to be assigned. Using a massively parallel supercomputer, we leverage the information already contained in the thousands of coded stories and are able to code a story in about 2 seconds.[1] Given SEEKER, the text retrieval system, we achieved these results in about two person-months. We believe this approach is effective in reducing the development time to implement classification systems involving large number of topics for the purpose of classification, message routing etc.

## 2 Introduction

Various successful systems have been developed to classify text documents including telegraphic messages [Young] [Goodman], physics abstracts [Biebricher], and full text news stories [Hayes] [Rau]. Some of the approaches rely on constructing topic definitions that require selection of relevant words and phrases or use case frames and other NLP techniques intended for more than classification e.g. for tasks such as extraction of relational information from text [Young] [Jacobs].

Alternative systems [Biebricher] [Lewis] use statistical approaches such as conditional probabilities on summary representations of the documents. One problem with statistical representations of the training database is the high dimensionality of the training space, generally at least 150k unique single features -- or words. Such a large feature space makes it difficult to compute probabilities involving conjunctions or co-occurrence of features. It also makes the application of neural networks a daunting task. We describe a new approach for classifying news stories using their full text that achieves high recall and at least moderate precision without requiring manual definitions of the various topics, as required by most of the earlier approaches.

Section 3 describes the problem; Section 4, the main results and Section 5 reviews MBR. The classification algorithm and variations of parameters are described in Sections 7 - 9 and we conclude with a discussion of results and future directions.

## 3 The News Story Classification Problem

Each day editors at Dow Jones assign codes to hundreds of stories originating from diverse sources such as newspapers, magazines, newswires, and press releases. Each editor must master the 350 or so distinct codes, grouped into seven categories: industry, market sector, product, subject, government agency, and region. (See Fig. 1 for examples from each category.) Due to the high volume of stories, typically several thousand per day, manually coding all stories consistently and with high recall in a timely manner is impractical. In general, different editors may code documents with varying levels of consistency, accuracy, and completeness.

The coding task consists of assigning one or more codes to a text document, from a possible set of about 350 codes. Fig. 2 shows the text of a typical story with codes. The codes appearing in the header are the ones assigned by the editors

[1] On a 4k CM-2 Connection Machine System.

* David Waltz is also affiliated with the Center for Complex Systems at Brandeis University, Waltham, MA, 02254.

15th Ann Int'l SIGIR '92/Denmark-6/92
© 1992 ACM 0-89791-524-0/92/0006/0059...$1.50

**FIGURE 1 Some Sample Codes**

| Code | Name | # of Documents |
|---|---|---|
| R/CA | California | 9811 |
| R/TX | Texas | 2813 |
| M/TEC | Technology | 9364 |
| M/FIN | Financial | 7264 |
| N/PDT | New Products/Services | 4149 |
| N/ERN | Earnings | 9841 |
| I/CPR | Computers | 2880 |
| I/BNK | All Banks | 2869 |
| P/CAR | Cars | 380 |
| P/PCR | Personal Computers | 315 |
| G/CNG | Congress | 307 |
| G/FDA | Food and Drug Admin. | 214 |

and the codes following "Suggested Codes" are those suggested by the automated system. Each code has a score in the left hand column, representing the contributions of several near matches. In this particular case the system suggests 11 of the 14 codes assigned by the editors (marked by *) and assigns three extra codes. By varying the score threshold, we can trade-off recall and precision.

## 4 Main Results

The table below groups performance by code category for a random test set of 1000 articles. The last column lists the different codes in each code category.

| Category | Name | Recall | Precision | # of Codes |
|---|---|---|---|---|
| I/ | industry | 91 | 85 | 112 |
| M/ | market sector | 93 | 91 | 9 |
| G/ | government | 85 | 87 | 28 |
| R/ | region | 86 | 64 | 121 |
| N/ | subject | 72 | 53 | 70 |
| P/ | product | 69 | 89 | 21 |
|  | Total | 81 | 70 | 361 |

Although the automated system achieves fair to high recall for all the code categories, consistent precision seems much harder. Given SEEKER, the text retrieval system as the underlying match engine, we achieved these results in about 2 person-months. By comparison, [Hayes] and [Creecy] report efforts of 2.5 and 8 person-years, respectively, for developing rule/pattern based concept descriptions for classification tasks with comparable numbers of categories. Our current speed of coding stories is about a story every 2 seconds on a 4k CM-2 system.

**FIGURE 2 Sample News Story and Codes**

0023000PR PR 910820
I/AUT I/CPR I/ELQ M/CYC M/IDU M/TEC
R/EU R/FE R/GE R/JA R/MI R/PRM R/TX R/WEU

Suggested Codes:

| | | |
|---|---|---|
| * 3991 | R/FE | Far East |
| * 3991 | M/IDU | Industrial |
| * 3991 | I/ELQ | Electrical Components & Equipment |
| * 3067 | R/JA | Japan |
| * 2813 | M/TEC | Technology |
| * 2813 | M/CYC | Consumer, Cyclical |
| * 2813 | I/CPR | Computers |
| * 2813 | I/AUT | Automobile Manufacturers |
| 2460 | P/MCR | Mainframes |
| 1555 | R/CA | California |
| 1495 | M/UTI | Utilities |
| *1285 | R/MI | Michigan |
| *1178 | R/PRM | Pacific Rim |
| *1175 | R/EU | Europe |

"DAIMLER-BENZ UNIT SIGNS $11,000,000 AGREEMENT FOR HITATCHI DATA SYSTEMS DISK DRIVES"

SANTA CLARA, Calif.--(BUSINESS WIRE)--Debis Systemhaus GmbH, a 100 percent subsidiary of Daimler-Benz, has signed a contract to purchase approximately $11 million (U.S.) of 7390 Disk Storage Subsystems. The 7390s will be installed in debis' data centers throughout Germany over the next 6 months.

Daimler-Benz is a diversified manufacturing and services company whose corporate units include Mercedes-Benz, AEG, Deutsche Aerospace and debis. Debis provides computing, communications and financial services along with insurance, trading and marketing services. The 7390 Disk Storage Subsystems are HDS' most advanced high-capacity storage subsystems capable of storing up to 22.7 gigabytes of data per cabinet. 22 gigabytes is the equivalent of approximately 15.7 million double-spaced typewritten pages. First shipped in October of 1990, the 7390s are used in conjunction with high-performance mainframe computers in a wide variety of businesses and enterprises.

Hitachi Data Systems is a joint venture company owned by Hitachi, Ltd. and Electronic Data Systems (EDS). The company markets a broad range of mainframe systems, peripheral products and services. Headquartered in Santa Clara, HDS employees 2,600 people with products installed in more than 30 countries worldwide.

## 5 The Memory Based Reasoning Approach

Memory Based Reasoning (MBR) consists of variations on the nearest neighbor techniques, (see [Dasrathy] for a comprehensive review of NN techniques). For a review of MBR see [Waltz and Stanfill] and [Waltz]. In its simplest formulation, MBR solves a new task by looking up examples of tasks similar to the new task and using similarity with these remembered solutions to determine the new solution. For example to assign occupation and industry codes to a new Census return one can look up near matches from a large (already coded) database and choose codes based on considering several near matches [Creecy]. In a similar fashion, codes are assigned to new unseen news stories by finding near matches from the training database and then choosing the best few codes based on a confidence threshold.

## 6 The Training Database

Dow Jones publishes a variety of news sources in electronic form. We used the source for press releases called PR Newswire, most of which is concerned with business news. Editors assign codes to stories daily. On average, a story has about 2,700 words and 8 codes. For the experiments reported here the training database consists of 49,652 examples (total size about 140 Mbytes). The database was not specially created for the project; it just contains stories from several months of the newswire. The training database has different numbers of stories for different codes and code categories. Figs. 1 and 3 show some representative codes and code categories and their sizes.

**FIGURE 3 Code Frequencies by Categories**

| Category | # of Documents | # of Occurrences |
|---|---|---|
| I/ | 38308 | 57430 |
| M/ | 38562 | 42058 |
| G/ | 3926 | 4200 |
| R/ | 47083 | 116358 |
| N/ | 41902 | 52751 |
| P/ | 2242 | 2523 |

## 7 The Classification Algorithm

Following the general approach of MBR, we first find the near matches for each document to be classified. This is done by constructing a relevance feedback query out of the text of the document, including both words and capitalized pairs. This query returns a weighted list of near matches (see Fig. 4). We assign codes to the unknown document by combining the codes assigned to the *k* nearest matches; for these experiments, we used up to 11 nearest neighbors. Codes are assigned weights by summing similarity scores from the near matches. Finally we choose the best codes based on a score threshold. Fig. 4 shows the headlines and the normalized scores for the example used in Fig. 2 and the first few near matches from the relevance feedback search.

**FIGURE 4 Sample News Story with Eleven Nearest Neighbors**

| Score | Size | Headline |
|---|---|---|
| 1000 | 2k | Daimler-Benz unit signs $11,000,000 agreement for Hitatchi Data |
| 924 | 2k | MCI signs agreement for Hitachi Data Systems disk drives |
| 654 | 2k | Delta Air Lines takes delivery of industry's first... |
| 631 | 2k | Crowley Maritime Corp. installs HDS EX |
| 607 | 2k | HDS announces 15 percent performance boost for EX Series processors |
| 604 | 2k | L.M. Ericsson installs two Hitachi Data Systems 420 mainframes |
| 571 | 2k | Gaz de France installs HDS EX 420 mainframe |
| 568 | 5k | Hitachi Data Systems announces two new models of EX Series mainframes |
| 568 | 2k | HDS announces ESA/390 schedule |
| 543 | 2k | SPRINT installs HDS EX 420 |
| 543 | 4k | Hitachi DataSystems announces new model of EX Series mainframes |
| 485 | 4k | HDS announces upgrades for installed 7490 subsystems |

## 8 Defining Features

Although MBR is conceptually simple, its implementation requires identifying features and associated metrics that enable easy and quantitative comparisons between different examples. A news story has a consistent structure: headline, author, date, main text, etc. Potentially one can use words and phrases and their co-occurrence from all these fields to create features [Creecy]. For the purpose of this project we used single words and capital word pairs as features, largely because SEEKER, the underlying document retrieval system used as a match engine, provides support for this functionality.

## 9 The Match Engine (SEEKER)

SEEKER is the production version of the text retrieval system reported in [Stanfill].[2] The text is compressed by eliminating stop words (368 non-content bearing words such as "the," "on," and "and") and then by eliminating the most common words that account for 20% of the occurrences in the database. The second step removes a total of 72 additional words. The remaining words, known as *searchable terms*, are assigned weights inversely proportional to their frequencies in the database. Although general phrases are ignored, pairs of capital words that occur more than once are recognized and are also searchable. There are over 250,000 searchable words and word pairs in this database. Relevance feedback is performed by constructing queries from all the text of the document. Response time for a retrieval request is under a second. All the work for this paper was done on a 4k CM-2 Connection Machine System.

## 10 Variation of Different Parameters

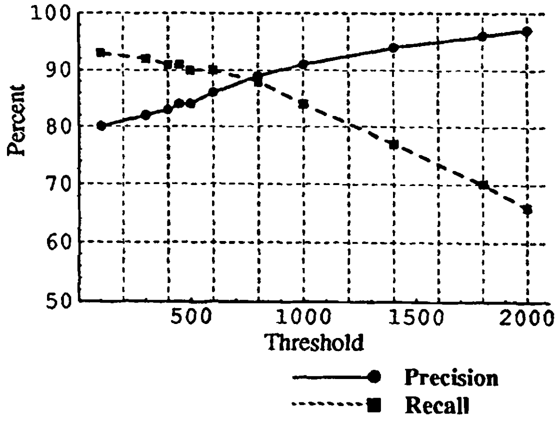
Different trade-offs between recall and precision can be achieved by varying the parameters of retrieval and classification. We describe the effects of varying the score threshold and $k$ the number of near matches used for classification.

### 10.1 Varying the confidence threshold

The following table (and Fig. 5) describes variation of recall and precision with respect to threshold for the I/ (industry) codes with $k = 10$.

| Threshold | Recall | Precision |
|---|---|---|
| 100 | 93 | 80 |
| 200 | 93 | 80 |
| 300 | 92 | 82 |
| 400 | 91 | 83 |
| 450 | 91 | 84 |
| 500 | 90 | 84 |
| 600 | 90 | 86 |
| 700 | 89 | 87 |
| 800 | 88 | 89 |
| 900 | 86 | 89 |
| 1000 | 84 | 91 |
| 1200 | 77 | 94 |
| 1500 | 70 | 96 |
| 2000 | 66 | 97 |

FIGURE 5 Variation by Threshold for Industry Codes, $k = 10$



### 10.2 Using different number of near matches

The following table (and Fig. 6) describes the variation of recall and precision for G/ (government) codes with threshold fixed at 100.

| Nearest-k | Recall | Precision |
|---|---|---|
| 1 | 57 | 93 |
| 2 | 69 | 90 |
| 3 | 72 | 88 |
| 4 | 76 | 87 |
| 5 | 79 | 84 |
| 6 | 79 | 83 |
| 7 | 84 | 86 |
| 8 | 85 | 86 |
| 9 | 86 | 87 |
| 10 | 85 | 87 |
| 11 | 86 | 85 |

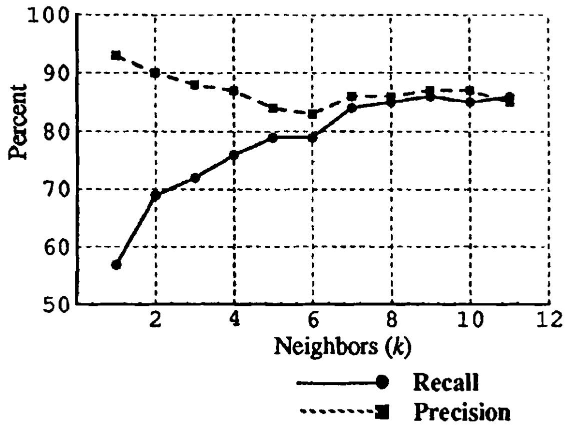
As expected, as the number of near matches considered increases, we find more correct codes but also add more noise. The optimal combination of score threshold and $k$ seems to differ depending on code categories and requires further study, possibly using more than eleven near matches.

### 10.3 Other parameters

One promising parameter is the weight for capital pairs. This would have the effect of increasing or decreasing classification relevance based on proper names (such as company, person and place names).

[2] Although the experiments were conducted at Thinking Machines Corp, a live version of the system is available from Dow Jones News Retrieval as *DowQuest*.

FIGURE 6 Variation by *k* for Government Codes, threshold = 100



## 11 Evaluating Performance

For the results reported in this paper we used n-way cross validation, which involves excluding each test example one at a time from the database and performing the classification on it. We used a randomly chosen set of 1000 articles for the test set.

In general we achieve better results on code categories with fewer numbers of codes. It is possible that more training examples for code categories with a larger number of codes would help improve performance.

The precision for subject codes (the N/ category) may be poor because the automatic system assigns more codes to a story than the 1 code per story assigned by the editors. It is also the category that requires most interpretation (since it classifies the type of event referred to by the story). Region codes (the R/ category) should perhaps be assigned by separate means (as are company codes) because the mere presence of a region term (such as the name of a state) may not imply that it is an important aspect for the story.

The weights used to determine similarity among documents are optimized for retrieval rather than classification. It is possible that word weights based on how many different categories the word appears in (e.g. sum of squared probabilities weights) or weights based on conditional probabilities might perform better [Creecy].

The performance reported here is the average for the entire database (as estimated by the test set). It should also be possible to define confidence levels for a document so that only documents with a high confidence would be classified automatically, with a high recall and precision, referring the difficult ones to the editors for manual coding.

### 11.1 Editorial evaluation

The results described in Section 4 are based on the assumption that the currently assigned codes are perfect i.e. that all existing codes are appropriate and no appropriate codes are missing from any documents. The extra codes assigned by the automatic system are judged as inappropriate. It is natural to ask how complete or consistent the original codes are.

In order to judge the relevance of the extra codes and also to assess the consistency of coding we asked the editors at Dow Jones to evaluate the codes assigned by the automatic system, as well as re-evaluate the original codes that were assigned to the documents. The results for the first phase of this evaluation are described next.

#### 11.1.1 Evaluation procedure

200 articles were selected at random. The articles were coded by the automatic system and the codes assigned were mixed with the original document codes, then sorted alphabetically to randomize them. These randomized codes (along with the text and without any scores) were evaluated by a single editor as *relevant* (correct), *irrelevant* (incorrect) and *borderline* (could be tolerated). The editor could also add for evaluation extra codes not on the list. The comparisons below summarizes the results. Due to the small size of the evaluation set (200 articles) from the point of view of statistical significance, the evaluation results are suggestive rather than definitive.

#### 11.1.2 Consistency of editorial coding

We compared the original document codes (assigned earlier by editors) to the most recent code evaluation. Treating the *relevant* category of the editorial evaluation as correct and excluding *borderline* codes as incorrect we find the following recall and precision for the original assigment:

Recall: 83%, Precision 88%

Including *borderline* codes as correct:

Recall: 61%, Precision 94%

This suggests that the editors are very consistent in their coding and that the *borderline* codes, the "maybe" category, are rarely assigned by the editors and should be treated as incorrect.

#### 11.1.3 Automatic coding vs. evaluated codes

Comparing the performance of the automatic system considering only the *relevant* codes as correct (excluding *borderline* codes as incorrect):

Recall: 80%, Precision 72%

which is about the same as compared to the original codes assigned to the documents. This is not a surprising result, given the high consistency of editorial coding.

Including *borderline* codes as correct:

Recall: 79%, Precision 73%

The automatic coding system does relatively well with respect to the *borderline* codes. However, it would seem better to filter them by improved confidence measures, since the editors do not often assign them.

## 12 Conclusions

We have demonstrated that a a relatively simple MBR approach enables news story classification with good recall and precision, for business-oriented news. While the performance seems less dramatic than certain systems that use manually constructed definitions (such as 90% recall and precision reported by [Hayes] and [Rau]) we believe that an MBR approach offers significant advantages in terms of ease of development, deployment, and maintenance. For instance entirely new codes can be added either by including stories with the new codes into the database or by adding the new codes to some earlier stories in the database.

We should be able to improve the performance by increasing the size of the training database since MBR systems benefit from larger databases [Creecy]. Our test database can hold more than 120,000 stories on the existing hardware.

Although we used an existing relevance feedback system as a match engine we believe it would be relatively easy to build a match engine for this specific purpose. This approach can also be used to provide classification at little extra cost where a news retrieval system with relevance feedback already exists.

The training database was created without having this application in mind; it constitutes several months of news stories, which were coded daily as part of the regular work of the editors. The application of MBR may also be relevant to other domains (such as OCR, patient records, financial assessments) where such coded free text databases are already available.

## 13 Future Work

We believe we can substantially improve the performance (especially precision) by optimizing the different parameters and combining the evidence from the nearest *k* neighbors in different ways (for instance by using different word weights). We would also like to study the performance with respect to the size of the database and its effect on code categories with different numbers of examples.

Since different sources may have style and content variations, it would be useful to see if a single training database can be used for different sources.

Adding automatically coded stories to the training database could cause a drift or bias in the coding process. Such a bias might improve performance by including new relevant documents in existing codes or might decrease performance if the new documents are irrelevant. We would like to quantify this effect.

## 14 Acknowledgments

We would like to thank Neil Budde and the Editorial Staff from Dow Jones and also Jonathan Goldman, Tracy Shen and Steve Smith from Thinking Machines for help with this project.

## 15 References

Biebricher, Peter; Fuhr, Norbert et al, "The Automatic Indexing System AIR/PHYS -- From Research to Application." Internal report, TH Darmstadt, Department of Computer Science, Darmstadt, Germany.

Creecy, R. H., Masand B., Smith S, Waltz D., "Trading MIPS and Memory for Knowledge Engineering: Classifying Census Returns on the Connection Machine." Forthcoming paper in the *Comm. ACM* (1992).

Dasrathy B. V. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* IEEE Computer Society Press, Los Alamitos, California (1991).

Goodman, M., "Prism, An AI Case Based Text Classification System." In R. Smith and E. Rappaport, (eds), *Innovative Applications of AI*, 1991, pp. 25 - 30.

Hayes, P. J. and Weinstein, S.P., "CONSTRUE/TIS: A System for Content-based Indexing of a Database of News Stories." *Innovative Applications of Artificial Intelligence 2*. The AAAI press/The MIT Press, Cambridge, Ma, pp. 49-64, 1991.

Hayes P.J. *et al*, "A Shell for Content-based Text Categorization." 6th IEEE AI Applications Conference, Santa Monica, March 1990.

Jacobs, Paul and Rau, Lisa. "SCISOR: Extracting Information from On-Line News." *Communications of the ACM*, 33(11):88-97, November 1990.

Lewis, David D., "An Evaluation of Phrasal and Clustered Representation on a Text Categorization Task." University of Chicago, personal communication, manuscript in progress.

Rau, Lisa F. and Jacobs, Paul S.,"Creating Segmented Databases From Free Text for Text Retrieval." *Proceedings, SIGIR 1991* (Chicago, Illinois).

Stanfill, C. and Kahle, B. "Parallel Free-Text Search on the Connection Machine System." *Comm. ACM* 29 *12* (December 1986), pp. 1229-1239.

Stanfill, C. and Waltz, D. L. "The Memory-Based Reasoning Paradigm." *Proc. Case-Based Reasoning Workshop*, Clearwater Beach, FL (May 1988), pp. 414-424.

Stanfill, C. and Waltz, D. L. "The Memory-Based Reasoning Paradigm." *Comm. ACM* 29 *12* (December 1986), pp. 1213-1228.

Young, Sheryl R., Hayes, Philip J., "Automatic Classification and Summarization of Banking Telexes." *Proceedings of the Second IEEE Conference on AI Applications*, 1985, Miami Beach, FL.

Waltz, D. L. "Memory-Based Reasoning." In M.A. Arbib and J.A. Robinson (eds), *Natural and Artificial Parallel Computation*, The MIT Press, Cambridge, Mass., (1990), pp. 251-276.