Effects of Query and Database Sizes on Classification of News Stories using Memory Based Reasoning

Brij Masand
Thinking Machines Corporation
245 First Street, Cambridge, Massachusetts, 02142 USA

Abstract

In this paper we explore the effects of query and database size on news story classification performance. Memory Based Reasoning (MBR) (a *k*-nearest neighbor method) is used as the classification method. There are 360 different possible codes. Close matches to a new story are found using an already coded training database of about 87,000 stories from the Dow Jones Press Release News Wire, and a Connection-Machine Document Retrieval system (CMDRS, [Stanfill]) that supports full text queries, as the underlying match engine. By combining the codes from the near matches, new stories are coded with a recall of about 80% and precision of about 70%, as reported in [Masand]. When the query size is varied from 10 terms to more than 200 terms (matched against the full-text of documents) the recall-precision product changes from 0.53 to 0.6. While this is a significant change, we find that moderate sized queries of 40-80 terms can suffice for finding relevant matches for classification. By changing the size of database from 10,000 stories to 80,000 we found that the recall-precision product changed from 0.22 to 0.57. This shows that with our current MBR approach the database size can't be reduced significantly without compromising performance. We also find that fewer number of retrieved matches are needed with larger query and database sizes.

1 Introduction

Finding near matches by designing appropriate metrics is an important step in the Case Based Reasoning (CBR) and the MBR paradigm. The availability of massively parallel machines with large memory and computation power have made it easier to use large example databases with relatively simple similarity metrics for retrieving relevant matches. In particular, increased computation power has made it easier to use all the features of a case as a query for matching against databases as large as several Gigabytes.

We have previously reported the results for using a CM - based IR system for classification of news stories [Masand] and the use of a large training database of Census returns for automatically classifying new Census returns[Creecy]. Both of these projects use large (more than 50,000) number of examples as training cases. The large size of the databases used have allowed fairly simple forms of indexing and metrics for assessing similarity. In this paper we report experiments to determine the optimal number of terms in a query for finding relevant news story matches for good classification performance. We also investigate how the classification performance scales with the size of the training database. The parameters of the classification system are optimized automatically for finding the best performance related to different query and database sizes.

Section 2 and 3 describe the coding problem and representative results. Section 4 reviews MBR and the details of the classification algorithm. The effect of different classification system parameters on performance and their automatic optimization is described in Section 5. Variations with respect to query and database size are described in Section 6. We conclude with a discussion of results and future directions.

2.0 Text Categorization: Related Work

Various successful systems have been developed to clas- sify text documents including telegraphic messages [Young] [Goodman], physics abstracts [Biebricher], and full text news stories [Hayes] [Lewis] [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 tasks more sophisticated 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. The case based telex routing work described by [Goodman] is closest in approach to our present work although it uses a much smaller case base combined with more post processing of the retrieved matches.

Using an MBR approach for classifying news stories using their full text we are able to achieve high recall and at least moderate precision without requiring manual definitions of the various topics, as required by most of the earlier approaches.

3.0 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.

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

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 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 (overlap marked by *) and assigns three extra codes. By varying the score threshold, we can trade-off recall and precision.

3.1 Representative Results

The table below groups performance by code category for a random test set of 200 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	82	74	121
N/	subject	70	76	70
P/	product	69	89	21
	Total	81	72	361

The automated system achieves fair to high recall and precision for all the code categories. Given CMDRS, the text retrieval system as the underlying match engine, these basic results were achieved 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.

4 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.

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.

From the CBR point of view the examples here are micro-cases with no annotated structures. This allows matching on entire examples including all the features (rather than a match based on a summary/indexed representation of an example or case). Partly because of the nature of the classification task and partly because of the excellent matches provided by using full-text matching, very little post-processing or "reasoning" is done to assign the codes, other than a straightforward summing of evidence for different codes from a number of near matches.

4.1 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 bytes (or 500 words) and 8 codes. For the experiments reported here the training database consists of 87,000 examples (total size about 240 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

4.2 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 full-text 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 normal-

ized 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

4.3 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 CMDRS, the underlying document retrieval system used as a match engine, provides support for this functionality. If the entire story is used as query, as is the case for CMDRS, this can result in a large query of several hundred terms even after ignoring stop words.

4.4 The Match Engine (CMDRS)

CMDRS is the production version of the text retrieval system reported in [Stanfill].[1] 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.

5 Optimization of parameters

Different trade-offs between recall and precision can be achieved by varying the parameters of retrieval and classification. Two important parameters are the number of near matches (k) used and the thresholds used for assigning the codes.The results in section 3.1 represent manual optimization of the number of near neighbors and the confidence thresholds. As the number of near matches considered for classification increases recall increases while precision decreases (more correct codes are found but also more noise is added). In general higher thresholds are effective when considering increasing number of (k) near neighbors.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.

Another parameter, the *code-gap-ratio*, measures the ratio between successive codes. By adjusting this parameter, the tail end of the assigned codes can be cutoff whenever there is a sharp drop in successive (ranked) scores. In order to reduce the effort required to find optimal combinations of the number of near neighbors, score and code-gap-ratio thresholds we experimented with automatic optimization through a random search.

5.1 Automated optimization

Before trying techniques such as hill-climbing we started by testing random n-tuples of parameters (we have experimented so far with up to 3 parameters) and choosing the best performance. To our surprise we found that in about a thousand random trials -- about half an hour's worth of computation -- optimal parameters are found for which the performance is quite close to or better than the values obtained by manual optimization. This is important both from the point of view of avoiding the cumbersome task of manual optimization as well as removing the subjective fac-

[1] 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 5 Variation of Classification performance by query size

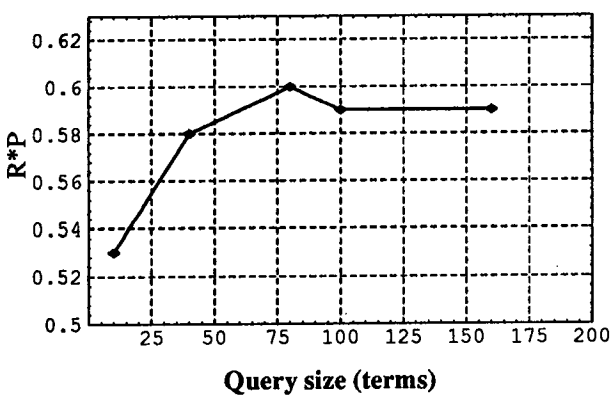
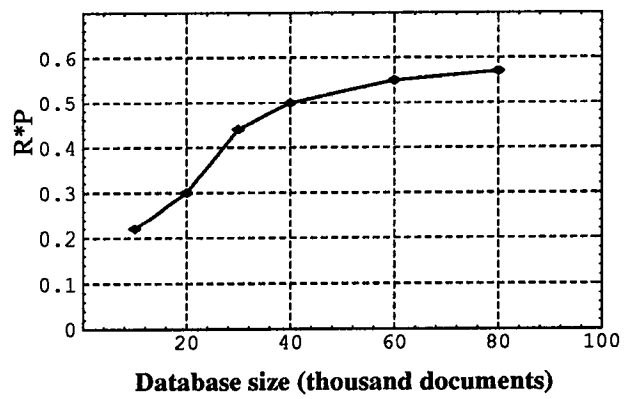


FIGURE 6 Variation of Classification performance by database size



tor when comparing the relative performance for varying database and query sizes. All the results in the following sections represent the outcome of automatic selection of optimal parameters through a simple random search.

6.0 Variation of Query and Database Sizes

6.1 Varying the query size

The following table and Fig. 5 describe the variation of the recall-precision product with respect to different number of terms in the query that is used to find the near matches. The query size represents the n best terms (based on inverse frequency weights) selected for the query from the article being classified.

Query Size (terms)	R*P	Recall	Precision	k
10	.53	68	79	7
40	.58	74	78	4
80	.6	73	82	4
100	.6	74	80	3
160	.6	72	82	3

We see that while the change is significant form 10 to 100 terms, a modest query size of 40 to 80 terms suffices for the best performance. The average size of a story in the database is about 500 words. Only the size of the query is varied, matching against the full-text of the documents. The terms consist of single words and pairs of adjacent capital words. The last column represents the average optimized number of near neighbors for a query size. Fewer matches seem to be needed as the query size increases, presumably because the queries are more specific as the size increases.

Different sets of near neighbors for the test set were found from the entire database, using different query sizes and then the best performance was found by automatically optimizing the number of near neighbors and associated thresholds.

6.2 Varying the database size

The following table and Fig. 6 describe the variation of recall-precision product with increasing database size.

Database size (thousands)	R*P	Recall	Precision	k
10	.22	42	52	5
20	.31	65	47	5
30	.44	62	71	3
40	.49	64	78	4
60	.55	69	80	4
80	.57	68	83	2

We see that performance improves significantly as stories are added in increments of 10,000. While the rate of increase seems to decrease, the data suggests that performance can be improved further by increasing the size of the database.

Different sets of near matches for the test set were found for different fractions of the database, while keeping the maximum query size constant to a few hundred terms. It's likely that there is relationship between the number of different classifications (in this case 361) and the rate of increase as the database is expanded.

The last column indicates the average size of the near matches for optimal performance. As the database increases fewer matches are needed, as one expects to find more number of close matches.

6.3 Other variations

One promising direction would be to explore the effect of using summary representation of documents themselves while still retaining a large training database. It would also be useful to see the interaction of query size with different database sizes and different number of codes to be assigned in various code categories.

7 Discussion

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 200 articles for the test set.

While we have shown that fairly small query sizes suffice to locate relevant near matches, one should note that the query is searched against the full text of the documents for which the total number of features exceed 250,000. In addition the smaller query sizes may benefit from a larger database since the probability of a close match increases with the size of the database. It's also possible that a large training database compensates for a simple similarity metric.

It would be interesting to see if automated optimization of parameters through a random search works for more than 3 parameters at one time. For larger sets of parameters, hill climbing or optimization through genetic algorithms might be necessary.

The increase of performance with increasing number of examples is probably different for different code-categories with different numbers of training examples.

8 Conclusions

We have shown that Recall and precision using the MBR approach for news story classification increased significantly when we increased the database from 10,000 to 87,0000 stories. Along with increased performance fewer number of retrieved matches seem to be needed for a larger database.

In addition we have shown that query sizes of about 40 to 80 terms (best terms as ranked by inverse frequency weights) are sufficient for locating the best relevant matches for the purpose of classification when matched against a full-text representation of the documents. Fewer matches seem to be needed for larger queries.

A simple random search yielded classification parameters that result in respectable performance, comparable to manual optimization.

While the automatically optimized 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 dramatically reduced time of development, automated optimization, ease of deployment, and maintenance. We should be able to improve the performance further by increasing the size of the training database. Our test database can hold more than 120,000 stories on the existing hardware (a 4k processor CM2).

This approach can be used to provide case based classification at little extra cost where a news retrieval system with relevance feedback already exists.

9 Future Work

Although we have shown that the performance of news classification with the MBR approach depends on having a large example-base it may be possible to prune the database by removing redundant examples and also removing examples that aren't often retrieved as matches.

We are studying the effect of scaling the database on different categories of codes with different numbers of codes in them, to judge the effect of training database size with respect to the granularity of classification.

We would like to study the combined effect of the variation of query size with respect to the scaling of the database as well as the effect of matching against a summary representation of the documents in the database. Another extension of the current approach might be to do more in-depth reasoning on the retrieved matches, as exemplified in the case-based reasoning approach.

10 Acknowledgments

I would like to thank Gordon Linoff, Dave Waltz and Steve Smith from Thinking Machines for help with this project.

11 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." *Comm. ACM* (August 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, CA, 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." *Proceedings, SIGIR '92*, Copenhagen,Denmark.

Masand, Brij M., Linoff, Gordon S.,and Waltz, David L., "Classifying News Stories on the Connection Machine using Memory Based Reasoning" *Proceedings, SIGIR '92*, Copenhagen, Denmark.

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, MA, (1990), pp. 251-276.