Automatic Essay Grading Using Text Categorization Techniques

Leah S. Larkey
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst
Amherst, MA 01003-4610, USA
www.cs.umass.edu/~larkey

Abstract Several standard text-categorization techniques were applied to the problem of automated essay grading. Bayesian independence classifiers and k-nearest-neighbor classifiers were trained to assign scores to manually-graded essays. These scores were combined with several other summary text measures using linear regression. The classifiers and regression equations were then applied to a new set of essays. The classifiers worked very well. The agreement between the automated grader and the final manual grade was as good as the agreement between human graders.

1 Introduction

Researchers have attempted to automate the grading of student essays since the 1960's [12]. The approach has been to define a large number of objectively measurable features in the essays, such as essay length, average word length, etc. and use multiple linear regression to try to predict the scores that human graders would give these essays. Even in this early work, results were surprisingly good. The scores assigned by computer correlated about .50 with the English teachers who provided the manually assigned grades. This was about as well as the English teachers correlated with each other. Another approach in more recent work has been to use a k-nearest-neighbor algorithm to access the k essays most similar to the new essay to be graded. The new essay receives a final score which is a weighted average of grades from those similar essays [5].

Now that students routinely produce essays by typing them directly into computers, computer grading is being revisited as a practical possibility. While there is a well justified resistance to letting the computer be the sole judge of the quality of student's work, it can play a useful auxiliary role. It could provide automated feedback for practice essays on the web. The computer grader could replace one human grader in cases where more than one judge typically does the grading, or could supplement the judgment of a classroom teacher who is the sole judge of the quality of an essay. A second human judge would then be needed only in those cases where the human and the computer grader disagree.

To be presented at the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval, Melbourne, Australia, 24–28 August 1998.

We are interested here in comparing a new and simple approach to computer grading of essays with the techniques tried previously. The new approach is to train binary classifiers to distinguish "good" from "bad" essays, and use the scores output by these classifiers to rank essays and assign grades to them. We aim to compare the binary classifier approach to the linear regression and k-nearest-neighbor approaches, and to see what can be gained by treating k-nearest-neighbor and binary classifier scores as additional variables in the linear regression.

A secondary goal of this study is to go beyond previous work by assigning a discrete grade to each essay, and by measuring exact agreement with the human raters. Previous work in this area has assigned continuous ranking scores to essays and used the Pearson product-moment correlation or r, between the human grader(s) and the computer grader as the criterial measure. The correlation does not indicate how often the computer grader would have assigned the correct grade. In fact, one need not even assign a grade to compute the correlation. The correlation is computed on the continuous ranking score. We assign an integral grade to each essay and measure the agreement between the computer grader and the human grader, as well as the correlation between the ranking score and the manually assigned grade.

2 The Experimental Data

We obtained five data sets from a large testing organization. Each essay in each set had been manually graded. The sets varied in the number of points in their grading scale and the size of the data sets. They covered widely different content areas and were aimed at different age groups. The first set, *Soc*, was a social studies question where certain facts were expected to be covered. The second set, *Phys*, was a physics question requiring an enumeration and discussion of different kinds of energy transformations in a particular situation. The third set, *Law*, required the evaluation of a legal argument presented in the question. The last two questions sets, *G1* and *G2*, were general questions from an exam for college students who want to pursue graduate studies. *G1* was a very general opinion question intended to evaluate how well the student could present a logical argument. *G2* presented a specific scenario with an argument the student had to evaluate. In our judgment, all the questions except *G1* required the student to cover certain points. In contrast, a good answer to *G1* would be judged less by what was covered than by how it was expressed.

For the first three sets, *Soc*, *Phys*, and *Law*, the only available manual score was based on an unknown number

	Train	Test	Grades
Soc	233	50	4
Phys	586	80	4
Law	223	50	7
G1	403	232	6
G2	383	225	6

Table 1: Data sets used in automatic essay grading experiments

of graders. The last two sets, *G1* and *G2* were manually scored by two graders. In addition to the scores assigned by the two graders, each essay was assigned a "final" score, which was usually, but not always, the average of the two graders' scores. Table 1 summarizes the characteristics of each data set. The columns headed *Train* and *Test* indicate the number of manually graded essays in each subset of documents for each type of essay. The column headed *Grades* indicates the number of points in the grading scale for that essay.

Normally in training involving parameters, one avoids overfitting by setting aside part of the training data as a tuning set, and chooses parameters and methods based on the results on the tuning set. However, preliminary work with different divisions of one of these data sets showed better results when all the training data were used in all phases of training. We speculate that this is due to the small size of all these data sets. It should be noted that we never used the test sets for any tuning or selection of parameters. All tuning was carried out on the training set.

3 Experiment 1

In experiment 1, we used the first three sets of essays, *Soc*, *Phys*, and *Law*. We trained Bayesian classifiers and k-nearest-neighbor classifiers and compared their performance with the linear regression approach using text-complexity features. Finally, we combined everything by using the two types of classifier outputs as variables in the linear regression along with the text-complexity features. In all cases, we derived thresholds to divide up the continuum of predicted scores into the appropriate number of each grade.

3.1 Bayesian Independence Classifiers

In experiment 1 we trained several binary Bayesian independence classifiers to distinguish better essays from worse essays, dividing the set at different points. For example, for essays graded on a four point scale, we trained a binary classifier to distinguish 1's from 2's, 3's, and 4's, another to distinguish 3's and 4's from 1's and 2's, and another to distinguish 4's from 1's, 2's, and 3's.

Bayesian independence classifiers, first proposed by Maron [10], are a type of general linear classifier (see the excellent overview in [9]). They estimate the probability that a document is a positive exemplar of a category, given the presence of certain words in the document. Fuhr [3] and Lewis [8] have explored improvements to Maron's model. Our model is similar to Lewis's, and has the following characteristics: First, a set of features (terms) is selected separately for each classifier. Bayes theorem is used to estimate the probability of category membership for each category and each document. Probability estimates are based on the co-occurrence of categories and the selected features in the training corpus, and some independence assumptions [2]. The two phases of training, feature selection and training of coefficients, were carried out in manner similar to that of [6], and are described more fully below.

Feature Selection First, occurrences of 418 stopwords were removed from the essays, and the remaining terms were stemmed using the *kstem* stemmer [4]. Any stemmed terms found in at least three essays in the positive training set were feature candidates. The selection of features from this set was carried out independently for each binary classifier as follows.

Expected mutual information (EMIM) [14] was computed for each feature, and the features were rank ordered by EMIM score. From this set, the final number of features chosen for the classifier was tuned on the training set of data. Classifier scores were computed for a range of feature set sizes, for each document in the training set. The feature set size which produced training document scores yielding the highest correlation with the manual scores was considered optimal.[1]

Coefficients The Bayesian classifier estimates the log probability that the essay belongs to the class of "good" documents, $\log P(C|Doc)$, as follows:

$$\log(P(C)) + \sum_i \begin{cases} \log \left(P(A_i|C)/P(A_i)\right) & \text{if the test doc has feature } A_i \\ \log \left(P(\overline{A}_i|C)/P(\overline{A}_i)\right) & \text{if the test doc does not have } A_i \end{cases}$$

Where $P(C)$ is the prior probability that any document is in class C, the class of "good" documents, $P(A_i|C)$ is the conditional probability of a document having feature A_i given that the document is in class C, $P(A_i)$ is the prior probability of any document containing feature A_i, $P(\overline{A}_i|C)$ is the conditional probability that a document does not have feature A_i given that the document is in class C, and $P(\overline{A}_i)$ is the prior probability that a document does not contain feature A_i. This is based on Lewis's binary model [7], which assigns 0's or 1's for feature weights depending upon whether terms are present or absent in a document. Binary values seemed adequate, because these essays are generally short, and even important terms are usually only mentioned once.

3.2 K-nearest-neighbor classifier

In k-nearest-neighbor classification we find the k essays in the training collection that are most similar to the test essay. The test essay then receives a score which is a similarity-weighted average of the grades that were manually assigned to the k retrieved training essays.

In our implementation, the similarity between a test essay and the training set was measured by the Inquery retrieval system, a probabilistic retrieval system using $tf \cdot idf$ weighting [1]. The entire test document was submitted as a query against a database of training documents. The resulting ranking score, or belief score, was used as the similarity metric. The parameter k, the number of top-ranked documents over which to average, was tuned on the training set. We chose the value that yielded the highest correlation with the manual ratings. This process yielded values of 45, 55, and 90, for the *Soc*, *Phys*, and *Law* essay sets.

[1] A criterion of average precision for the binary classifier yielded very similar results.

3.3 Text-Complexity Features

The following eleven features were used to characterize each document:

- The number of characters in the document (*Chars*)
- The number of words in the document (*Words*)
- The number of different words in the document (*Diffwds*)
- The fourth root of the number of words in the document (Page [12] finds this to be a useful predictive variable.) (*Rootwds*)
- The number of sentences in the document (*Sents*)
- Average word length (*Wordlen=Chars/Words*)
- Average sentence length (*Sentlen=Words/Sents*)
- Number of words longer than 5 characters (*BW5*)
- Number of words longer than 6 characters (*BW6*)
- Number of words longer than 7 characters (*BW7*)
- Number of words longer than 8 characters (*BW8*)

3.4 Linear Regression

The SPSS stepwise linear regression package was used to select those variables which accounted for the highest variance in the data, and to compute coefficients for them. We performed regressions using three different combinations of variables: (1) the eleven text-complexity variables (2) just the Bayesian classifiers, and (3) all the variables - the eleven text-complexity complexity variables, the k-nearest-neighbor score, and the scores output by the Bayesian classifiers.

3.5 Thresholds

Using the regression equation derived from the training data, we computed a predicted score for each training essay, and rank-ordered the essays by that score. Category cutoffs were chosen to put the correct number of training essays into each grade. This technique is known as *proportional assignment* [7]. These cutoff scores were then used to determine the assignment of grades from scores in the test set. For the individual classifiers, cutoff scores were derived the same way, but based on the k-nearest-neighbor and Bayesian classifier scores rather than on a regression score.

3.6 Measures

For this first experiment, we report three different measures to capture the extent to which grades were assigned correctly: the Pearson product-moment correlation (r) and two other measures of interest to testing agencies, the proportion of cases where the same score was assigned (*Exact*) and the proportion of cases where the score assigned was at most one point away from the correct score (*Adjacent*). Unlike the correlation, these measures capture how much one scoring procedure actually agrees with another scoring procedure. We are particularly interested in comparing our algorithm's performance on these three measures with the two human graders. We have the individual judges' grades only for the last two data sets, *G1* and *G2*, which we report in Experiment 2.

Variable	Exact	Adjacent	r	Components
Text	.56	.94	.73	BW6, Rootwds Wordlen
Knn (45)	.54	.96	.69	
B1 (200)	.58	.94	.71	
B2 (180)	.66	1.00	.77	
B3 (140)	.60	1.00	.77	
B4 (240)	.62	.98	.78	
All Bayes	.62	1.00	.78	B2, B3
All	.60	1.00	.77	Sents, B2,B3

Table 2: Results on *Soc* data set

Variable	Exact	Adjacent	r	Components
Text	.47	.91	.56	Sents, Wordlen Rootwds
Knn (55)	.44	.90	.53	
B1 (320)	.51	.90	.61	
B2 (480)	.50	.89	.59	
B3 (420)	.55	.90	.63	
B4 (240)	.49	.89	.61	B1,B3
All Bayes	.50	.89	.63	B1,B3
All	.47	.93	.59	B2,B3,B4 BW7,Diffwds, Wordlen Rootwds

Table 3: Results on *Phys* data set

3.7 Results

Tables 2 through 4 show the results on the first three data sets, *Soc*, *Phys*, and *Law*. The column labeled *Variable* indicates which variable or group of variables contributed to the score. *Text* indicates the linear regression involving the text-complexity variables listed in Section 3.3. *Knn* indicates the k-nearest-neighbor classifier alone. $B1$, $B2$, etc. indicate the individual Bayesian classifiers trained on different partitions of the training essays into "good" and "bad." *All Bayes* indicates the composite grader based on linear regression of the Bayesian classifiers. *All* is the grader based on the linear regression using all the available variables. When there is a number included in parentheses next to the variable name, it shows the value of the parameter set for that variable. For *Knn*, that parameter is the number of training documents that contribute to the score. For the Bayesian classifiers, the parameter is the number of terms included in the classifier. The columns labeled *Exact*, *Adjacent*, and r show the measures described in Section 3.6. The column labeled *Components* show the variables that the stepwise linear regression included in the regression equation for that combination of variables. Only conditions involving linear regression have an entry in this column.

Performance on the *Soc* data set appears very good, but the *Phys* set is less so. Both were graded on a four point scale, yet all three measures are consistently lower on the *Phys* data set. Performance on the *Law* set is also quite good. Although the Exact and Adjacent scores are lower than on the *Soc* data set, one would expect this on a seven-point scale compared to a four-point scale. The correlations are in roughly the same range. Some generalizations can be made across all three data sets, despite the differences in level of performance. First, the Bayesian independence classifiers performed better

Variable	Exact	Adjacent	r	Components
Text	.24	.66	.57	Rootwds
Knn(90)	.40	.66	.61	
B1 (50)	.36	.54	.60	
B2 (120)	.32	.72	.75	
B3 (300)	.28	.72	.74	
B4 (300)	.28	.84	.76	
B5 (120)	.36	.82	.76	
B6 (160)	.42	.86	.79	
B7 (160)	.32	.78	.78	
All Bayes	.32	.84	.79	B2,B3,B6
All	.36	.84	.77	B2,B3,B6, Knn,BW6

Table 4: Results on *Law* data set

Variable	Exact	Adjacent	r	Components
Text	.51	.94	.86	Diffwds, Sents, BW6
Knn (220)	.42	.84	.75	
B1 (300)	.36	.82	.69	
B2 (320)	.47	.95	.84	
B3 (300)	.48	.94	.84	
B4 (280)	.47	.92	.83	
B5 (380)	.47	.94	.82	
B6 (600)	.50	.96	.86	
All Bayes	.50	.96	.86	B1,B2,B5,B6
All	.55	.97	.88	B1,B5,B6,BW5 BW6, Sents Rootwds,Knn

Table 5: Results on *G1* data set

Variable	Exact	Adjacent	r	Components
Text	.42	.92	.83	BW5
Knn (180)	.34	.84	.77	
B1 (600)	.36	.86	.77	
B2 (320)	.48	.95	.85	
B3 (300)	.46	.96	.86	
B4 (280)	.52	.95	.85	
B5 (300)	.48	.95	.85	
B6 (680)	.48	.95	.84	
All Bayes	.52	.96	.86	B1,B3,B5
All	.52	.96	.88	B1,B3,B5, BW8, Diffwds, Rootwds

Table 6: Results on *G2* data set

than the text-complexity variables alone or the k-nearest-neighbor classifier. In the *Text* condition, *Rootwds*, the fourth root of essay length, was always selected as one of the variables. In the *All* condition, in which all available variables were in the regression, the length variables were not as obviously important. Two of the three sets included a word length variable (*Wordlen*, *BW6*, *BW7*) and two of the three sets included an essay length variable (*Sents*, *Diffwds*, *Rootwds*). In the *All* condition at least two Bayesian classifiers were always selected, but the k-nearest-neighbor score was selected for only one of the three data sets. Finally, the performance of the final regression equation (*All*) was not consistently better than the performance using the regression-selected Bayesian classifiers (*All Bayes*).

3.8 Discussion of Experiment 1

The performance of these various algorithms on automatic essay grading is varied. Performance on the *Soc* data set seems very good, although it is hard to judge how good it should be. It is striking that a certain fairly consistent level of performance was achieved using the Bayesian classifiers, and that adding text complexity features and k-nearest-neighbor scores did not appear to produce much better performance. The additional variables did improve performance on the training data, which is why they were included, but the improvement did not always hold on the independent test data. These different variables seem to measure the same underlying properties of the data, so beyond a certain minimal coverage, addition of new variables adds only redundant information. This impression is confirmed by an examination of the correlation matrix containing all the variables that went into the regression equation.

Our results differ from previous work, which always found some kind of essay length variable to be extremely important. In [12], a large proportion of the variance was always accounted for by the fourth root of the essay length, and in [5], a vector length variable was very important. In contrast, our results only found length variables to be prominent when Bayesian classifiers were not included in the regression. In all three data sets, the regression selected *Rootwds*, the fourth root of the essay length in words, as an important variable when only text complexity variables were included. In contrast, when Bayesian classifiers were included in the regression equation, at least two Bayesian classifiers were always selected, and length variables were not consistently selected. We speculate that our Bayesian classifiers captured the same variance in the data. An essay that received a high scores from a Bayesian classifier would contain a large number of terms with positive weights for that classifier, and would thus have to be long enough to contain that large number of terms.

4 Experiment 2

In experiment two, we used the last two data sets, *G1* and *G2*. For these sets we had grades assigned by two separate human judges, as well as the final grade given to each essay. This allowed us to compare the level of agreement we obtained between the automatic grading and the final grade with the level of agreement found between the two human graders. This comparison makes the absolute levels of performance more interpretable than in Experiment 1. The training procedure was the same as in Experiment 1.

4.1 Experiment 2 Results

Table 5 and Table 6 summarize the results on the last two data sets. The results on *G2* are completely consistent with Experiment 1. Bayesian classifiers were superior to text-complexity and k-nearest-neighbor methods. The combination of all classifiers was at best only slightly better than the combination of Bayesian classifiers. On *G1*, the exception to the pattern was that the text-complexity variables alone performed as well as the Bayesian classifiers. The combination classifier was superior to all the others, particularly in the Exact score.

	Exact	Adjacent	r
G1: auto vs. manual(final)	.55	.97	.88
G1: manual A vs. B	.56	.95	.87
G2: auto vs. manual(final)	.52	.96	.86
G2: manual A vs. B	.56	.95	.88

Table 7: Comparison with Human Graders

4.2 Comparison with Human Graders

Table 7 shows the agreement between the final manually assigned grades and the grade automatically assigned by the combination *All*. For comparison, the agreement between the two human graders is also shown. The numbers are very close.

5 Discussion

Automated essay grading works surprisingly well. Correlations are generally in the high .70's and .80's, depending upon essay type and presumably upon the quality of the human ratings. These levels are comparable to those attained by Landauer et al. [5] and Page [12]. These correlations seem high, and are comparable to the correlations between human judges.

For the *Exact* and *Adjacent* measures, our algorithms found the "correct" grade around 50-65% of the time on the four and six point rating scales, and were within one point of the "correct" grade 90-100% of the time. This is about the same as the agreement between the two human judges on *G1* and *G2*. Other studies do not report actual agreement between the automatic graders and humans, nor do they mention computing a discrete grade at all.

Previous work, particularly by Page [12], has had great success with text-complexity variables like those listed in section 3.3. We found these variables to be adequate only for one of the five data sets, *G1*. *G1* was the only opinion question in the group. We speculate that in this type of question, the fluency with which ideas are expressed may be more important than the content of those ideas. However, some of Page's variables were more sophisticated than ours, for example those involving a measure of how successfully a parsing algorithm could parse the essay. It is possible the use of more sophisticated text-complexity measures would have improved the performance.

We were surprised to find that the tuning of our Bayesian classifiers preferred so many features. The usual guidelines are to have a ratio of 5 to 10 training samples per features, though others recommend having as many as 50 to 100 [7]. We used as many as 680 features in some of our classifiers, which seemed large, so we did some additional post hoc analyses to see how the test results varied with this parameter. On the training data, variations in number of features yielded quite small changes in the correlations between the binary classifier scores and the grade, except at the extreme low end. These variations produced larger differences in the test data. In fact, the tuning on the training data did choose roughly the best performing classifiers for the test data. It might have made more sense to tune the number of features on a separate set of data, but there were not enough essays in this set to separate the training data into two parts. Given that the large number of features really was improving the classifiers, why would this be so?

Normally a classifier is doing the job of inferring whether a document is about something or relevant to something. One expects the core of a category to be characterized by a few key concepts, and perhaps some larger number of highly associated concepts. The job of feature selection is to find these. In contrast, in essay grading, the classifier is trying to determine whether an essay is "good" or not. This kind of judgment depends on the exhaustiveness with which a topic is treated, and it can be treated many different ways, hence a very large number of different features can contribute to the "goodness" of an essay.

This large number of terms in the binary classifiers is a likely explanation of why essay length variables were not found to be as important as in other studies of essay grading. Length variables are summary measures of how many words, or how many different words are used in an essay. The scores on our binary classifiers are summary measures that capture how many words are used in the essay which are differentially associated with good essays. These scores would be highly correlated with length, but would probably be better than length in cases where a successful essay must cover a specific set of concepts.

Another interesting outcome of the parameter tuning on these data was the high value of k found for the k-nearest-neighbor classifier. In previous work done by us and others using k-nearest-neighbor classification for text, values of k on the order of 10 to 30 were found to be optimal [6, 15, 11, 13]. We were surprised to find much higher values of k to be optimal for essay grading. A reasonable explanation may be the following. In other studies, the k-nearest-neighbor classifier tries to find the small subset of documents in a database that are in the same class (or classes) as the test document. This subset allows one to find the appropriate class for the test document from the classes assigned to the retrieved documents. The essay grading case differs, however, in that all the documents are about the same topic as the test document, so the grade assigned to any similar document has something to contribute to the grade of the test essay.

This work showed the k-nearest-neighbor approach to be distinctly inferior to both the other approaches. Recently, Landauer, et al. [5] have applied Latent Semantic Analysis in a k-nearest-neighbor approach to the problem of essay grading. They got very good results, which suggests that the use of more sophisticated features or a different similarity metric may work better.

In conclusion, binary classifiers which attempted to separate "good" from "bad" essays produced a successful automated essay grader. The evidence suggests that many different approaches can produce about the same level of performance.

Acknowledgements This material is based on work supported in part by the National Science Foundation, Library of Congress, and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings, and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor. I would like to thank Scott Elliot for helping us obtain the data and for his many suggestions.

References

[1] J.P. Callan, W.B. Croft, and J. Broglio. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, pages 327–343, 1995.

[2] William S. Cooper. Some inconsistencies and misnomers in probabilistic information retrieval. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 57–61, 1991.

[3] Norbert Fuhr. Models for retrieval with probabilistic indexing. *Information Processing and Management*, 25(1):55–72, 1989.

[4] Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 191–203, 1993.

[5] T. Landauer, D. Laham, B. Rehder, and M. Schreiner. How well can passage meaning be derived without using word order? A comparison of latent semantic analysis and humans. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, 1997.

[6] Leah S. Larkey and W. Bruce Croft. Combining classifiers in text categorization. In *Proceedings of the 19^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 289–298, 1996.

[7] David Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50, 1992.

[8] David D. Lewis. *Representation and Learning in Information Retrieval*. PhD thesis, University of Massachusetts, 1992.

[9] David D. Lewis, Robert E. Shapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In *Proceedings of the 19^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

[10] M.E. Maron. Automatic indexing: An experimental inquiry. *Journal of the Association for Computing Machinery*, 8:404–417, 1961.

[11] Brij Masand, Gordon Linoff, and David Waltz. Classifying news stories using memory based reasoning. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59–65, 1992.

[12] Ellis B. Page. Computer grading of student prose, using modern concepts and software. *Journal of Experimental Education*, 62(2):127–142, 1994.

[13] C. Stanfill and David Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, December 1986.

[14] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.

[15] Yiming Yang and Christopher G. Chute. An application of Expert Network to clinical classification and MEDLINE indexing. In *Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care*, pages 157–161, 1994.