An evaluation of self-prediction models for adaptive text-categorization

Richard Forsyth[1], Shaaron Ainsworth[1] and David Clarke[1].

[1]**School of Psychology, University of Nottingham, NG7 2RD.**

Correspondence: forsyth_rich@yahoo.co.uk

Abstract

Despite an ever-increasing abundance of extralinguistic electronic information (audio files, video recordings, event logfiles etc.), data in the form of text is still a fundamental resource for researchers in a wide variety of fields. Social scientists, for example, frequently spend a great deal of time and effort recording, transcribing, segmenting, and categorizing segments of natural language. Each segment may often be only a few words in length. Natural-language text, whether written language or transcribed talk, is gathered in situations as diverse as interviews, online communities, email, face to face conversations, and chatrooms. Researchers in many disciplines use such data for a variety of purposes, typically after assigning categorical codes to short stretches of their texts.

The present paper reports on the progress on the CodeLearner project, which applies machine-learning techniques to assist in the categorical coding of short text segments. We focus particularly on two issues raised by the attempt to automate this analytical process: firstly, the inescapably iterative nature of the process of training a computer system to emulate an analyst's coding decisions; secondly (and consequently) the fact that such **a system must predict its own future performance** in order for this iterative process to terminate without wasted effort. Given that most coding schemes are generated and/or refined during analysis, adding self-prediction to the learning system is a necessity: it enables the researcher to re-train it at any convenient interval -- for instance, after coding every twenty or fifty new instances -- and have it estimate its own accuracy at various future points. This allows early exit from problems where the learning system proves incapable of the categorization task or finely-judged effort

in situations where it is capable of reaching a sufficient accuracy level. In either case, human effort is saved.

Empirical trials on five text corpora suggest that a simple three-parameter model enables the computer system to predict its own future "learning curve" with sufficient accuracy to make iterative computer-assisted coding a practical proposition. It is noted that success in this enterprise will have methodological implications: it will make feasible an incremental, exploratory approach to machine-assisted coding, thus opening up areas of research where the coding task would previously have been dismissed as too arduous.

Keywords: Active learning, Applied linguistics, Bayesian methods, linguistic computing, machine learning, Markov models, qualitative computing, self-prediction, text classification.

1. Introduction

In linguistics and many branches of the social sciences, researchers must spend a great deal of time and effort coding or annotating segments of text. A number of researchers have therefore proposed employing machine-learning methods to ease this task (e.g. Daelmans & van den Bosch, 2005; Donmez et al., 2005). However, existing machine-learning tools tend to work on the assumption that the user has a sufficiently large training set of instances which have been pre-categorized with "gold standard" category labels. In other words, they are not very well suited to the situation of the

researcher who has a novel or non-standard coding scheme (typical in the social sciences) and wants to code as few instances as possible and then let the learning system classify the remainder, but has no way of estimating how many instances will have to be coded to reach a satisfactory performance level.

This issue has been addressed within the computational linguistics community under the banner of "active learning" (Mackay, D., 1992; Cohn et al., 1996; Jones et al., 2003; Becker et al., 2005). In an active-learning approach the human expert gives category labels to a batch of instances and then learning system chooses which of the unlabelled cases should be in the next batch to be coded by hand -- this procedure being repeated until a criterion level is reached. In some problem domains this kind of example selection can have a dramatic effect. Becker et al. (2005) give an example in the field of named-entity recognition where such a systematic selection of instances to be coded by hand would result either in a reduction of 38.5% in the number of cases to be coded compared to random selection or a reduction of 13% in the error-rate if the same number of cases are coded.

Active learning thus allows an incremental approach to machine-assisted text categorization. What it doesn't attempt to do is predict how many instances will need to be hand-coded before a given criterion level of performance is reached. In particular, it does not provide a mechanism for allowing a user to quit as soon as possible in the not unusual situation where no amount (or no reasonable amount) of training data will allow the learning system to reach an acceptable level of accuracy. To do so would require that the system estimate, in advance, its performance level

when any given number of additional training instances have been coded. This function, which we term *self-prediction*, is the focus of the present study.

In the context of computer-assisted categorical coding of text segments, it is important that the system be able to predict from a relatively small batch of hand-coded examples what its likely future performance will be on a much larger set. This is because the user of such a system will typically operate within a decision-making loop of the following general nature.

- code a batch of text segments manually;
- test the supervised learning system on the cases coded so far (using a form of resampling such as cross-validation to avoid biased error estimates);
- decide which of the following options to take:
    - **stop**, since the desired accuracy level has been obtained;
    - **continue** to code more training cases because the required accuracy can be reached with more training data;
    - **abandon** the attempt since the accuracy criterion will never be reached or reached only with excessive effort.

The rest of this paper describes an empirical evaluation of three different methods of self-prediction. In section 2 we describe the five corpora used in this evaluation study. Section 3 outlines the learning algorithm employed. Section 4 describes the three self-prediction models tested. Results are presented in section 5 and discussed in section 6.

2. Trial data sets

Our five trial data sets are intended to represent a cross-section of the kinds of categorical coding tasks that arise with natural-language data. Brief descriptions of them are given in Table 1.

Table 1. The five trial data sets. [about here]

Our initial dataset, the Cardiac learners' data, reflects our concern with human learning, and in particular the effect of self-explanation on learning. Self-explanations are pieces of knowledge generated by an individual learner that state something which is not explicit in the information they are learning from (Chi et al., 1989). This is of interest because learners develop a deeper understanding of the material they are studying if they give more self-explanations. A study was conducted in which learners studied either abstract or realistic diagrams (Ainsworth et al., 2007) of the human cardio-vascular system. While doing so, they were encouraged to verbalize their reactions to the materials presented. The transcripts were segmented and coded according to the nature of the utterance (paraphrase, self-explanation statement, or monitoring statement). The learning system's task is to replicate this three-way classification.

The second dataset consists of a selection of texts by Alexander Hamilton and James Madison, the two main authors of the *Federalist* papers, which were published in 1787-1788 and have never been out of print since. These essays gave rise to a

celebrated and difficult case of disputed authorship which was subject to a ground-breaking stylometric analysis by Mosteller & Wallace (1984 [1964]) and which has become an accepted benchmark in the field of authorship attribution. Further details can be found in Holmes & Forsyth (1995). For the present investigation it should be noted that only **undisputed** papers by the two authors were chosen, 17 by Hamilton and 14 by Madison. In addition, two state of the union addresses given by Madison when he was president (in 1811 and 1813) were added to make the amount of text by both authors more nearly balanced. Note also that the unit of analysis in this problem was the paragraph: the classification program's task is to classify individual paragraphs according to their author, not whole documents. Clearly classifying paragraphs (mean size 145 words) is a more challenging task than classifying entire documents (mean size over 2500 words). Even so the average length of text segments to be classified in this problem was the longest of our five data sets.

The third dataset consists of a total of 1810 (fictional) utterances from sixteen novels by Agatha Christie. Eight of these novels featured Jane Marple and eight featured Hercule Poirot as the lead detective. All quoted items of dialogue ascribed to these two personages were extracted and saved in 16 separate files. Then the words "Jane", "Miss" and "Hercule" were replaced by "X" while Marple and Poirot were replaced by "Y". (Miss Marple is in fact seldom referred to as "Jane" by her creator.) The task of the learning system is to assign each saying to its putative speaker.

The fourth dataset comes from an unpublished report by one of the present authors (Forsyth, 2004) and consists of free-form comments made by respondents in a survey designed to evaluate a management training programme. These free-text responses

(average length less than eight words) were categorized by hand into one of 14 thematic categories, of which the first four were AO, CC, DT and GD -- standing for Administration & Organization, Course Content, Duration & Timing, and Group Dynamics, respectively. Two examples are: "subject matter varied from expectations" (CC), and "input from other delegates helped" (GD). The program's task is to assign each short text segment to its correct thematic category. This is intended to represent the kind of problem that arises in political or consumer surveys, for example. Success in this enterprise would have important practical implications in market research and allied fields.

The fifth dataset is the Maptask corpus (Anderson et al., 1991) which is publicly available from the Human Communication Research Centre, Edinburgh. This corpus consists of 128 dialogues generated by students at Glasgow who were undertaking the Map Task (Brown et al., 1984). In the Map Task one participant, the instruction giver, has a sketch map with a route marked on it and the other participant, the instruction follower, has a map without this route. The instruction follower has to draw a path reproducing the route, as accurately as possible, on his or her map. Neither party can see the other's map, and the maps have a small number of divergences which necessitate certain navigational negotiations. The resultant conversations have been transcribed and divided into a total of 27,084 *moves*, each of which is assigned to one of 13 dialogue-act categories according to its pragmatic function within the discourse, namely: Acknowledge, Align, Check, Clarify, Explain, Instruct, Query-w, Query-yn, Ready, Reply-n, Reply-w, Reply-y or Uncodable. For more information on the dialogue acts, see Carletta et al. (1997). The program's task is to assign each move to

its appropriate class simply on the basis of the text constituting that move (on average less than six words).

Thus these five data sets represent a variety of text-classification tasks of practical importance. Information about the sizes of these data sets is shown in Table 2.

Table 2. Characteristics of the five data sets. [about here]

3. Learning algorithm

A large number of algorithms has been used for text classification (e.g. Yang, 1999; Stamatatos et al., 2001; Sebastiani, 2002; Peng et al., 2003). The algorithm we chose for our initial explorations is essentially a generalization of that described in Khmelev & Tweedie (2001), which has been shown to give good results in the area of authorship attribution in both English and Russian. This algorithm -- which itself is a variant of the widely-used Naive Bayes Classifier, as described, for instance, in Mitchell (1997) -- creates a simple Markovian model of the language in the training dataset and uses Bayesian inference to arrived at probabilistic category assignments (on training or test data). Hence we term our generalization of it a *Bayes-Markov Classifier* (BMC).

The advantages of this simple and robust algorithm include the following: it requires no pre-processing step to select features (in effect, all features are used); it requires no external support software, such as taggers, or lexicons; and it could potentially be applied to languages other than English (though in this paper we report only trials on

English texts). Moreover, it employs a Bayesian inferential framework, which has served as the basis for several practical text-categorization systems, such as spam filtering (Sahami et al., 1998). However, we do not wish to claim that this algorithm is the best possible for this purpose, only that it achieves acceptable accuracy levels, since the major focus of this investigation is the ability of the system in which the classifier is embedded to forecast its own future performance (see section 4). We plan to test alternative classification algorithms in due course.

3.1 The Bayes-Markov classifier

The system is as described by Khmelev & Tweedie (2001) with two extensions:

1) their system used character bigrams (pairs) as the basis for its language model, whereas ours permits n-grams of any length and allows word-based as well as character-based n-grams, and is thus more flexible;
2) their system simply ignored attributes with a zero frequency in the training data, whereas ours uses the so-called "m-estimate" procedure (see, for instance, Cestnik & Bratko, 1991) which has the side-effect of attenuating extreme probabilities, including zero and one; hence no attributes are completely ignored.

Our algorithm is also very similar to that of Peng et al. (2003), the only differences being that ours uses a different smoothing technique (item (2) above) and that it allows words as well as characters to be the basic units.

The terminology for n-gram length is not consistent in the literature. In the present paper n refers to *order* of the underlying Markovian model (cf. McMahon & Smith, 1998), or equivalently the length of the prefix or context in which the probability of a particular event (character or word) is being estimated. For example, in character mode with a n-gram length of 2, and a string "pho" the system would use the conditional probability

P("o" | "ph")

i.e. the probability of finding an "o" immediately after the bigram "ph" as the basis for its computations. (Some researchers would call this an n-gram size of 3, as it involves three successive tokens, e.g. Oakes (1998: 243).)

For the experiments described below, the distinction between upper and lower case letters was ignored.

3.2 Operational framework

From our point of view, the operational framework is more important than the classification algorithm it contains. In this case, the framework is a program, written in Python, that acts as a test harness. This repetitively selects random subsamples, of increasing sizes, from the full data set as training sets and selects disjoint random subsamples, of fixed user-specified size, as testing sets. This enables analysis of the algorithm's performance as the number of training instances increases, and permits the plotting of "learning curves" which illustrate the relationship between the amount of training data and the accuracy of classification (on unseen hold-out data).

We do not simply specify the accuracy of the classifier using (almost) the whole data set, as is typically done using the method of n-fold cross-validation, because we also want to investigate whether the system can predict its own success rate on (almost) the whole data set (or indeed on any size of sample) from a smaller subsample.

The behaviour of the test harness program is outlined in pseudo-code below.

```
Obtain test-set size (N2) from user
For a user-specified number of repetitions
        Set initial training-set size to N1 [normally zero]
        While training-set size plus test-set size (N1+N2) does not exceed data set size
                Pick a random subsample of size N1 as training set
                Pick a disjoint random subsample of size N2 as testing set
                Train classifier (e.g. BMC) on training data
                Test classifier on testing data
                Record results
                Increment N1 by user-specified increment (N3)
        Append results to file for subsequent analyses
```

When the size of the training sample (N1) is zero, the classifier simply makes a random category choice with probability 1/K, where K is the number of categories. This baseline behaviour corresponds to complete ignorance, where the classifier does not even have information on the relative frequency of the categories.

4. Prediction models

It seems natural, when seeking ways of predicting the behaviour of a learning system, to begin searching in those disciplines that concern themselves with adaptive or self-improving natural systems, especially animate organisms or commercial entities.

Learning curves of various kinds have been studied since the time of Ebbinghaus (1913 [1885]). In Psychology, during the heyday of behaviourism, a number of attempts were made to fit simple mathematical equations to data derived from experiments on humans, pigeons, rats or other animate agents. Of these, perhaps the most influential has been that of Clark Hull (1943) which takes the form of an exponential formula

$$Y = a + b * (1 - 10 ^ (c*x))$$

where Y is a performance index (in Hull's terms, evidence of the theoretical construct habit strength) and x is the number of learning trials that the organism has been subjected to. The coefficients a, b and c are parameters to be estimated from data. (Coefficient a can be omitted if the initial performance level is zero, so a 2-parameter version is often quoted, but a zero intercept cannot be assumed in our case, where guesswork gives a non-zero success rate.) This formula has been criticized and the theory behind it amended in various ways, most notably by Rescorla and Wagner (1972), but even modern versions are "based on the assumption that the rate of growth is proportional to the amount of growth still possible" (Bolles, 1979: 94) which implies that the mathematical structure remains an exponential form.

A separate tradition of modelling learning curves (also called "experience curves") has developed in the field of Management Science, originating from an article by Wright (1936). In this context a typical application is predicting the labour cost or materials cost per unit of a given product as the number of units manufactured increases. This is found to decrease systematically as the factory or firm gains experience of making the product. Wright proposed a power law of the form

$$Y = a + b * x \wedge c$$

where Y is the output measure, such as cost per unit manufactured, and x is the total number of units manufactured. A number of alternative formulae have been proposed (see Yelle, 1979) but by far the most popular remains Wright's original power law (e.g. Nahmias, 2004).

It should be noted that whether a learning curve slopes upwards or down depends on whether a cost index, such as effort or time, or a success index, such as proportion of correct responses, is being measured. The direction, however, does not change the structure of the model, only the sign(s) of certain coefficients.

To address the issue of how well a learning system such as ours could predict its own performance, we decided to compare these different formulae -- one deriving from the work of Wright (1936) as typically applied in business and economics, one deriving from the work of Clark Hull (1943) as developed in Psychology. In addition, a third formula, also using three parameters, was composed for the present purpose. This is a

simplification of the double log-inverse formula used in some branches of econometrics, e.g. by Saxon (1975) to forecast food consumption as a function of income.

$$Y = a + b * \ln(x+1) + c * 1/(x+1)$$

In our variant of this formula, Y is the result (success rate) and x is the training-set size; ln() is the natural logarithm.

The three models under consideration are listed in Table 3.

Table 3. Three models of the "learning curve". [about here]

All three formulae have equivalent degrees of freedom in the sense of having three adjustable parameters. These parameters were optimized by using the non-linear regression method nls() in R (Crawley, 2002), and their quality was assessed by the mean squared error (MSE) criterion.

5. Results

The system was run on all five trial data sets in 2 modes, word mode with n-gram size of 1 (W1) and character mode with an n-gram size of 2 (C2). Pilot studies have shown that these n-gram sizes were usually good choices for the two different modes. Thus the three prediction formulae were compared ten times. Table 4 gives details of the experimental runs.

Table 4. Details of experimental trials. [about here]

In each case the step size was chosen to give 20 equally-spaced points along the range of training-set sizes at which the system was trained and tested. The training and test sets were always disjoint randomly chosen subsamples of the full dataset.

5.1 Accuracy of classification

Although endpoint accuracy of the classifier is not our primary concern in this paper, for completeness Table 5 summarizes the results on the five test problems by giving for each of the datasets the percentage of correct decisions made on the test set by the BMC when trained on the largest training-set size for that dataset.

Table 5. Classification accuracy (% correct decisions) on trial datasets. [about here]

The column showing the most frequent category in the dataset is included mainly to show that the BMC is always comfortably superior to the strategy of guessing the most frequent category label. The final column, expected chance success rate is calculated according to a formula given in Klecka (1980) derived from that originated by Mosteller & Bush (1954), i.e. the product of the proportions in each category. This is even more comfortably exceeded by the actual success rates.

The kinds of relationship found between size of training set and classification accuracy are illustrated in Figures 1 and 2 below. Figure 1 shows a rapid initial rise

followed by early deceleration towards a plateau. Figure 2 shows a slower but steadier rise.

Figure 1. Boxplot illustrating learning system's progress on Cardiac dataset. [about here]

Figure 2. Boxplot illustrating learning system's progress on Federalist Data. [about here]

All "learning curves" have similar overall shapes, but with important differences in slope along their lengths. In general, it is likely that simpler methods will show faster learning in the early stages but slower improvement as the amount of training data is increased. Thus, in order to help the user decide how many examples need to be expertly encoded, it becomes important for the system to model the shape of its own learning curve.

5.2 Accuracy of self-prediction (50/50 split)

There are several ways that the accuracy of a self-prediction model could be assessed and since this is a novel field no accepted standard method exists. In the present investigation the following procedure was adopted. First, for each of the five datasets, a split-point was made at the nearest training-set size to half the size of the full

dataset. Next the parameters of all three models were optimized, using the nls() non-linear regression function (Crawley, 2002) of the statistical package R, on the accuracy results for all training-set sizes up to the split-point. Then the three formulae were applied to predict the accuracy results of all training-set sizes greater than the split-point. This enabled **interpolation** accuracy to be measured by the mean of the squared errors (MSEs) between the predicted and the actual values up to the split-point, and **extrapolation** accuracy to be measured by the mean of squared errors (MSEs) between predicted and actual values beyond (greater than) the split-point.

The rationale for starting with a split point at half-way is that an equal split makes fewest a priori assumptions about the relative balance between two competing desiderata: having as large an interpolation set as possible to enable accurate parameter estimation, and having as large an extrapolation set as possible to gain a realistic assessment of self-prediction in the same context as the envisaged practical application, i.e. predicting the learning system's future accuracy on a larger training sample given a smaller one.

Figure 3 illustrates the curve-fitting and assessment process graphically using the Fictecs dataset.

Figure 3. Interpolation and extrapolation on Fictecs data: dotted line = Exponential model; dashed line = Power law; solid line = Log-inverse formula. [about here]

In this diagram, the open circles represent classification accuracy scores achieved by the learning system on test data over a range of training-set sizes. In this case there are 100 replications at each training-set size. The solid line depicts the relationship between the y-values (percentage correct decisions on test data) and the x-values (size of training sample) according to the Log-inverse formula. The dashed line represents the same relationship as predicted by the Power-law formula. The dotted line is the same relationship as predicted by the Exponential formula. The gap in the lines near the midpoint shows where **interpolation** ends and **extrapolation** begins, with interpolation on the left and extrapolation on the right.

Inspection of the right-hand (extrapolated) half of this graph reveals what might be called a "Goldilocks effect"[1]: the Power-law model makes forecasts that are somewhat too high; the Exponential model makes forecasts that are too low; while the Log-inverse formula gives forecasts in between the other two. This general pattern, with the Log-inverse model giving "middling" forecasts, is found in all ten trials.

Table 6 summarizes the results of these trials (five datasets in two modes). This table gives the mean squared deviations between the models' predicted (extrapolated) values and the true accuracy values for all three models on all ten trials. Higher MSE scores indicate worse predictions.

Table 6. Mean squared error scores (MSEs) between extrapolated and actual accuracies (50/50 split). [about here]

It is evident from these figures that the Exponential model performs poorly. Of the three models, it has the highest MSE score (i.e. is the least accurate) in every single case. This is doubtless because the Exponential learning-curve formula has a hard asymptotic maximum, whereas the other two models allow continued growth, albeit decelerating. By contrast, the Log-inverse model has the lowest (best) scores in nine of the ten trials.

The Exponential formula can be eliminated as a contender for the best of these three self-prediction models on the basis of these results. As for the other two models, the mean difference between the deviation scores of the Power and Log-inverse formulae (last 2 columns in Table 6) is 3.61, in favour of the Log-inverse model. To test whether this difference is significant, a paired t-test (2-tailed) was carried out. This gave a statistically significant result ($t = 2.55$, $df = 9$, $p = 0.03$). This was confirmed by a non-parametric test (Wilcoxon signed rank test, $V = 52$, $p = 0.0144$). To the extent that these datasets are representative this indicates that the Log-inverse formula is to be preferred for this kind of self-prediction task.

5.3 Accuracy of self-prediction (25/75 split)

However, since the practical application of these self-prediction formulae would be under circumstances where the size of the extrapolated dataset would be substantially larger than the data available for interpolation, a further test was performed, also using MSEs but with the split made at the 25% point. Thus the quality of the extrapolation is being assessed on data sizes up to four times as large as the largest size available for the parameter optimization. This is clearly a more stringent, and arguably a more realistic, test.

Table 7 gives figures for the extrapolation accuracy of all three models on all five data sets in both modes with a 25/75 split between interpolation and extrapolation. This is the same information as given in Table 6 for the 50/50 split.

Table 7. MSE scores between extrapolated and actual accuracies (25/75 split). [about here]

Once again the Exponential model performs poorly, having the worst error score nine times out of ten, while the Log-inverse model performs relatively well, having the best error score eight times out of ten.

These figures show that a 25/75 split does, as expected, pose a more severe test than a 50/50 split: in only 2 of the cells of Table 7 are the MSEs smaller (better) than the corresponding scores in Table 6. The variability of the scores has also increased. For each of the three columns the variance is greater for the 25/75 than the 50/50 split. However, a variance-ratio test (two-tailed) reveals that only for the first two models is this increase significant. (For the Exponential model, $F(9,9) = 0.1732$, $p = 0.0155$; for the Power-law model, $F(9,9) = 0.1104$, $p = 0.0030$; for the Log-inverse model, $F(9,9) = 0.5799$, $p = 0.4294$.)

This increase in variability entails a greater likelihood that the extrapolations will go seriously astray, as illustrated by the example of Figure 4.

Figure 4. Interpolation and extrapolation on Federalist data: dotted line = Exponential model; dashed line = Power law; solid line = Log-inverse formula. [about here]

In Figure 4, the gap between interpolation and extrapolation comes between training-set sizes 125 and 150. Here it can clearly be seen that the Power-law prediction formula has seriously over-estimated the learning system's future accuracy.

Once again, the Log-inverse formula's extrapolations fall in between those of the other two models. In fact this "Goldilocks" pattern, with the Log-inverse extrapolations being intermediate, is again repeated for all ten trials.

5.4 Further findings

In order to carry out more detailed analyses, the data were first re-scaled. The need for this re-scaling arises because the raw errors scores for the five different data sets are very different. This can be seen by comparing, for instance, the MSE scores for the Freetext with the Maptask data in either Table 6 or Table 7. To render the scores comparable over the five data sets, each MSE score for extrapolation was divided by the mean of the interpolation MSEs for the (three) models with the same mode on the same dataset. These re-scaled data are ratio scores, and because MSEs are error scores ratios greater than 1 signify that the extrapolation is less accurate than interpolation for the same dataset (using the same unit mode) and ratios less than 1 signify that the extrapolation is more accurate than interpolation for the given dataset and unit mode.

Using these rescaled data, a Bartlett test for homogeneity of variances was performed on both 50/50-split and 25/75-split extrapolations. In both cases this revealed

significant differences between the three groups: for 50/50-split, Bartlett's K-squared = 32.02, df = 2, $p < 0.00000001$; for 25/75 split, Bartlett's K-squared = 18.242, df = 2, $p = 0.0001$). Thus, not only does variability increase as the size of the interpolation set goes down from 50% to 25% of the maximum (as indicated in the previous section) but the three models differ in the variability of their scores. This precludes a straightforward parametric analysis, such as Analysis of Variance.

The next step, therefore, was to check whether the two unit modes (C2 and W1) differed significantly in terms of their MSE ratios. This was done by subtracting the MSE ratio for each extrapolation in W1 mode from the corresponding MSE ratio in C2 mode (same dataset, same formula) and applying a non-parametric Wilcoxon signed rank test to the differences (n=30). The median of these differences was 1.2 (favouring the C2 mode slightly) but this was not statistically significant ($V = 287$, p-value = 0.271). This was taken to imply that the analysis could proceed by lumping the two unit modes together.

Having thus, in effect, removed the effect of differences in difficulty between data sets, and having found that differences between unit modes could be ignored, the effects of the two main factors under study in this investigation (model type and size of interpolation set) can be summed up visually in a single image, Figure 5, below.

Figure 5. Mean MSE ratios for extrapolations, plus or minus 1 standard error, for all three models in both interpolation conditions. [about here]

This shows visually that:

- the extrapolation error gets larger as the amount of interpolation data is reduced;
- its variability increases for each model as the amount of interpolation data is reduced;
- the variability increases least for the Log-inverse model;
- the variability of the Exponential model is highest of the three in both interpolation conditions;
- the MSE ratio (error score) of the Log-inverse model is lowest of the three in both conditions;
- the MSE ratio of the Log-inverse model increases least of the three methods (worsens least) as the interpolation set size is reduced from 50% to 25%.

As parametric analysis was deemed inappropriate (because of the inhomogeneity of variances, noted above) a Kruskal-Wallis non-parametric comparison between the three models was performed for both interpolation conditions separately. In both cases the three groups of MSE ratio scores were found to differ significantly: for 50/50-split, Kruskal-Wallis chi-squared = 16.88, df = 2, p-value = 0.0002161; for 25/75-split, Kruskal-Wallis chi-squared = 14.59, df = 2, p-value = 0.0006803. This confirms the visual impression from Figure 5 that the three models have different MSE ratios.

Since it was obvious from the raw data that the Exponential model was a poor choice for this application, a separate 2-group comparison was also performed between the MSE ratio scores for the Power-law and the Log-inverse models, which are not so obviously differentiated in the diagram. Here the difference was not statistically

significant with the 50/50 split (Wilcoxon rank-sum test W = 27, p-value = 0.08921) but it was significantly different for the 25/75 split (Wilcoxon rank sum test W = 23, p-value = 0.04326).

This implies that in the more realistic, and more stringent, 25/75 test the Log-inverse formula's extrapolations are more accurate than those of the Power-law formula (as well as being less variable). Given the foregoing results, if a "winner" of this contest had to be declared it could only be the Log-inverse formula.

A subsidiary, perhaps counter-intuitive, finding arising from this comparison of median MSE ratios is summarized in table 8.

Table 8. Median MSE ratio for all three models in both interpolation conditions. [about here]

This shows that, on average, the mean squared error of the Exponential and Power-law models is higher (worse) when extrapolating than interpolating. Anyone familiar with machine learning will find this unsurprising: it corresponds to the common experience of finding higher error rates on test data than on training data. However, with the Log-inverse model, the position is reversed. It is actually more accurate, on average, when extrapolating than interpolating.

This is a curiosity which, if repeated on a larger trial with a wider range of datasets, would require explanation. It may simply be artefactual, arising because the rate of change in slope of the second part of a learning curve is almost inevitably less than

the first part. For the present, it can be taken as reassurance that the Log-inverse formula, if used in practice, could quote its self-predictions with confidence intervals that are unlikely to be optimistically biased -- a desirable characteristic.

MSE has certain desirable characteristics as a basis for error minimization, but since it is a squared measure it is not easily interpretable in terms of the problem domain; so as a more interpretable quality score the mean absolute deviations (MADs) between the Log-inverse predictions and the true values in the 25/75 condition were calculated at the highest testing size for each dataset in both unit modes. These ranged from 1.101 for the Maptask data in word mode to 5.741 for the Freetext data in character mode, with a mean of 3.507. To put this in a practical context, when this model is given a certain number of cases for interpolation its expected discrepancy in extrapolating to the learning system's success rate on a dataset four times as large is about 3.5 percentage points. Whether this is an acceptable level of discrepancy is problem-dependent, but it seems to us that discrepancies of this order would not be unreasonable in many practical applications.

Another finding perhaps worthy of remark emerged from this experiment: there was a significant (negative) rank correlation between the size of the dataset and the error scores for all three models. Thus extrapolations based on larger datasets tend to be more accurate. This is only to be expected, but it means that even the best of these formulae will not perform very well on small samples.

6. Discussion

We do not claim that the Bayes-Markov Classifier (BMC) is a definitive solution to the problem of categorical coding of short text segments. Nevertheless these initial results seem promising, especially in view of the fact that the algorithm has access only to the textual content of each short segment separately, with no contextual information.

As far as self-prediction is concerned, the Log-inverse formula enabled the system to predict its own future performance with greater accuracy and lower variability than two alternative models from different academic disciplines, both backed by a respectable body of research and practice developed over more than 60 years. This formula is doubtless not the best that can be found. However, the ease with which acceptable self-prediction can be achieved suggests that the reason for its rarity in learning systems is not that it is inherently difficult but that the need for it has not been recognized. This in turn means that addressing the needs of users with novel coding schemes within an iterative machine-learning loop is feasible, thus opening opportunities for numerous practical applications that would otherwise be considered infeasible.

The scope of this study is limited, so more remains to be done. An obvious future line of investigation is to test a wider range of self-prediction formulae. However, since dozens of formulae with potential application to the present problem can be found in various scientific literatures, and hundreds more could be concocted ad hoc, this could be a never-ending labour. It may be best, if such an avenue is followed, to employ an

evolutionary-computing methodology such as Genetic Programming (Koza, 1992). This would involve specifying the atomic elements (functions, operators and variables) from which a formula could be constructed, as well as a fitness function, and allowing an evolutionary system to **evolve** a most-suited formula using the technique known as symbolic regression.

Such an approach would finesse the need for a full and representative benchmark set of test corpora. We have chosen five data sets that we believe to cover a reasonable range of practical problem types, but we recognize that such benchmark collections can always be improved. With symbolic regression, however, as long as the evolutionary optimization method was integrated into the learning software, the generation and optimization of the self-prediction function could be done on the current user's data. There would be no need to find a "winning" formula over a standard set of test problems: each user would have a bespoke formula suited to his or her own data. (However, implementing such a system would require further research as well as substantial software-engineering resources, so a simple yet robust formula such as the Log-inverse will remain a valuable tool for some time to come.)

Another future prospect worth considering is the integration of self-prediction with active learning. The two method share a common motivation but they are orthogonal in the sense that any particular implementation of a trainable text-categorization system could incorporate either or both or neither. In the long run, we envisage that the state of the art will be to employ both methods.

Acknowledgements

This work was partly supported by Grant R149230035 of the UK Economic and Social Research Council as part of the DReSS Project sponsored by the National Centre for e-Social Science and partly by the School of Psychology, University of Nottingham. We would also like to thank Claire O'Malley of the School of Psychology for valuable guidance on this project, as well as Ryan Baker of Carnegie-Mellon University and Phoenix Lam of the Hong Kong Polytechnic University for helpful comments on an earlier draft of this paper.

Notes

[1] For readers unfamiliar with the fable of "Goldilocks and the three Bears", Goldilocks is a little lost girl who stumbles upon an empty house in the forest where she finds several groups of household items in threes, for example, three bowls of porridge. When she tests one bowl it is too hot, another is too cold, but the third is "just right".

References:

Ainsworth, S.E., Forsyth, R.S., Clarke, D.D., Robertson, L. & O'Malley, C. (2007). Automatic coding of learner's self-explanations when learning from diagrams. *EARLI 2007 Conference*, Szeged, Hungary, 28 August - 1 Sept 2007.

Anderson, A.H., Bader, M., Bard, E., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C. & Thompson, H. (1991). The HCRC Map Task corpus. *Language & Speech*, 34, 351-360.

[http://www.hcrc.ed.ac.uk/maptask]

Becker, M., Hachey, B., Alex, B. & Grover C. (2005). Optimising selective sampling for bootstrapping named entity recognition. *Proc. of Workshop on Learning with Multiple Views*, ICML, Bonn, 2005.

Bolles, R.C. (1979). *Learning theory*, second edition. Harcourt Brace Jovanovich, Orlando, Florida.

Brown, G., Anderson, A.H., Yule, G. & Shillcock, R.. (1984). *Teaching Talk*. Cambridge: Cambridge University Press.

Carletta, J., Isard, A., Isard, S., Kowtko, J.C., Doherty-Sneddon, G., Anderson, A.H. (1997). The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1), 13-31.

Cestnik, B. & Bratko, I. (1991). On estimating probabilities in tree pruning. *Fifth European Working Session on Learning*, EWSL 91, Porto, Portugal, Springer-Verlag.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 5, 145-182.

Cohn, D.A., Ghahramani, Z. & Jordan, M.I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4, 129-145.

Crawley, M.J. (2002). Statistical Computing: an introduction to data analysis using S-Plus. Chichester: John Wiley & Sons Ltd.

Daelemans, W. & van den Bosch, A. (2005). *Memory-based language processing*. Cambridge University Press, Cambridge.

Donmez, P., Rosé, C. P., Stegmann, K., Weinberger, A., and Fischer, F. (2005). Supporting CSCL with Automatic Corpus Analysis Technology, *Proceedings of Computer Supported Collaborative Learning***.**

Ebbinghaus, H. (1913 [1885]). *Memory: A Contribution to Experimental Psychology.* Translated by H.A. Ruger & C.E. Bussenius (1913). New York: Teachers College, Columbia University.

[http://psychclassics.yorku.ca/Ebbinghaus/index.htm, accessed 6/9/2007]

Forsyth, R.S. (2004). *Leadership skills for managers: evaluation report.* Unpublished report for the European Social Fund, University of Luton.

Hamilton, A., Madison, J. & Jay, J. (1992 [1788]). *The Federalist Papers*. London: Dent. (ed.) W.R. Brock.

Holmes, D.I. & Forsyth, R.S. (1995). The 'Federalist' revisited: new directions in authorship attribution. *Literary & Linguistic Computing*, 10(2), 111-127.

Hull, C.L. (1943). *Principles of Behavior*. New York: Appleton-Century-Crofts.

Jones, R., Ghani, R., Mitchell, T. & Riloff, E. (2003). Active learning with multiple view feature sets. *ECML 2003 Workshop on Adaptive Text Extraction and Mining*.

Khmelev, D.V. & Tweedie, F.J. (2001). Using Markov chains for identification of writers. *Literary & Linguistic Computing*, 16(3), 299-307.

Klecka, W.R. (1980). *Discriminant analysis*. Sage Publications, Newbury Park, California.

Koza, J. (1992). *Genetic programming*. Cambridge, MA: MIT Press.

Mackay, D. Information-based objective functions for active data selection. *Neural Computation*, 4(4), 590-604.

McMahon, J. & Smith, F.J. (1998). A review of statistical language processing techniques. *Artificial Intelligence Review*, 12, 347-391.

Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill, New York.

Mosteller, F. & Bush, R.R. (1954). Selected quantitative techniques. In G. Lindzey (ed.) *The Handbook of Social Psychology*, Vol.1, Addison-Wesley, Reading, Mass. 289-334.

Mosteller, F. & Wallace, D.L. (1984 [1964]). *Applied Bayesian and Classical Inference: the Case of the Federalist papers*. New York: Springer-Verlag. [Extended edition of *Inference and Disputed Authorship*. Reading, Mass: Addison Wesley (1964).]

Nahmias, S. (2004). *Production and Operations Analysis*, fifth edition. McGraw-Hill, Boston, Mass.

Oakes, M.P. (1998). *Statistics for Corpus Linguistics*. Edinburgh University Press.

Peng, F., Schuurmans, D., Keselj, V. & Wang, S.. (2003). Language independent authorship attribution using character level language models. *European Association of Computational Linguistics*, EACL2003, Budapest, 12-17 April, 2003.

Rescorla, R.A. & Wagner, A.R. (1972). A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In A. Black & W.F. Prokasy (eds.) *Classical conditioning: II Current Research and Theory*. New York: Appleton-Century-Crofts.

Sahami, M., Dumais, S., Heckerman, D. & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *AAAI98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, 27 July 1998.

Saxon, E.A. (1975). *Food Consumption in Japan*. Occasional paper no. 32, Bureau of Agricultural Economics. Canberra: Australian Government Publishing Service.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1-47.

Stamatatos, E., Fakotakis, N. & Kokkinakis, G, (2001). Automatic text categorization in terms of genre and author. Computational Linguistics, 26(4), 471-495.

Wright, T.P. (1936). Factors affecting the cost of airplanes. *J. Aeronautical Sciences*, 3(4), 122-128.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1, 69-90.

Yelle, L.E. (1979). The learning curve: historical review and comprehensive survey. *Decision Sciences*, 10, 302-328.

Tables (1-8)

Table 1. The five trial data sets.

Dataset	**Description**	**Mode**	**Task type**
Cardiac: Monologues of learners studying circulatory system (Ainsworth et al., 2007)	Verbalizations by learners studying text or diagrams explaining the workings of the human blood-circulation system	Spoken monologues	Functional coding
Fedpaps: Federalist papers (Hamilton et al., 1992 [1788])	Political texts written by Alexander Hamilton (n=17) or James Madison (n=16) [Undisputed essays only]	Written	Authorship attribution
Fictecs: Dialogue segments attributed to fictional detectives	Quoted speech by either Jane Marple or Hercule Poirot from 16 detective novels written by Agatha Christie	Written	Stylistic discrimination
Freetext: Free-form responses from user-feedback questionnaires	Free-form text responses in post-course student satisfaction survey	Written	Thematic categorization
Maptask: Map task dialogues (Anderson et al., 1991)	Conversations between Scottish students working on the Map Task (Brown et al., 1984)	Spoken dialogues	Dialogue-act classification

Table 2. Characteristics of the five data sets.

Dataset	**Size (words)**	**Segments**	**Mean segment length (words)**	**Categories**
Cardiac	23,330	1784	13.1	3
Fedpaps	84,594	583	145.1	2
Fictecs	50754	1810	28.0	2
Freetext	2083	264	7.9	14
Maptask	156310	27084	5.8	13

Table 3. Three models of the "learning curve".

Formula	**Origin**	**Type**	**Typical Application**
y ~ a + b * x ^ c	Wright (1936)	Power	Business: predicting cost per unit as number of units manufactured increases
y ~ a + b * (1 - 10 ^ (c * x))	Hull (1943)	Exponential	Psychology: predicting error rate as number of learning trials increases
y ~ a + b * ln(x+1) + c * 1/(x+1)	Saxon (1975)	Log-inverse	Econometrics: predicting consumption as income increases

Table 4. Details of experimental trials.

Dataset	**Text segments in full dataset**	**Range of training-set sizes**	**Step size**	**Testing set size (unseen)**	**Repetitions at each training-set size**
Cardiac	1784	0 .. 1680	84	100	100
Fedpaps	583	0 .. 500	25	80	100
Fictecs	1810	0 .. 1700	85	100	100
Freetext	264	0 .. 200	10	64	100
Maptask	27084	0 .. 26000	1300	1000	50

Table 5. Classification accuracy (% correct decisions) on trial datasets.

Dataset	**Segments / Categories**	**Mean accuracy (SD) in character mode, C2**	**Mean accuracy (SD) in word mode, W1**	**Percentage frequency of most common category**	**Percentage success expected by chance**
Cardiac	1784 / 3	74.07 (4.81)	70.05 (4.06)	57.29	48.28
Fedpaps	583 / 2	80.09 (4.04)	84.15 (3.83)	55.57	50.62
Fictecs	1810 / 2	88.25 (3.05)	83.53 (3.90)	58.84	51.56
Freetext	264 / 14	41.11 (5.57)	28.03 (4.67)	10.98	8.73
Maptask	27084 / 13	60.40 (1.43)	60.40 (1.37)	20.69	11.39

Table 6. Mean squared error scores (MSEs) between extrapolated and actual accuracies (50/50 split).

Dataset	**Mode**	**Exponential formula**	**Power-law formula**	**Log-inverse formula**
Cardiac Learners	C2	26.71	23.18	21.86
	W1	37.00	26.55	21.54
Federalist Essays	C2	29.03	33.13	19.96
	W1	31.38	26.07	17.01
Fictional Detectives	C2	23.68	16.80	11.79
	W1	21.83	15.14	13.81
Freetext Responses	C2	50.28	33.68	32.35
	W1	26.11	25.56	25.98
Maptask Dialogues	C2	5.86	2.70	2.57
	W1	11.37	2.29	2.14

Table 7. MSE scores between extrapolated and actual accuracies (25/75 split).

Dataset	**Mode**	**Exponential formula**	**Power-law formula**	**Log-inverse formula**
Cardiac Learners	C2	33.84	31.97	25.86
	W1	58.83	48.57	24.27
Federalist Essays	C2	39.96	120.95	23.99
	W1	75.23	34.46	19.03
Fictional Detectives	C2	49.59	27.43	13.37
	W1	35.46	15.44	14.08
Freetext Responses	C2	113.94	35.60	40.40
	W1	37.95	27.83	36.00
Maptask Dialogues	C2	9.60	2.94	2.70
	W1	19.32	2.13	2.08

Table 8. Median MSE ratio for all three models in both interpolation conditions.

	Exponential Model	**Power-law Model**	**Log-inverse Model**
50/50-split condition	1.2226	1.0076	0.8212
25/75-split condition	2.3609	1.3239	0.9178

Figures (1-5)

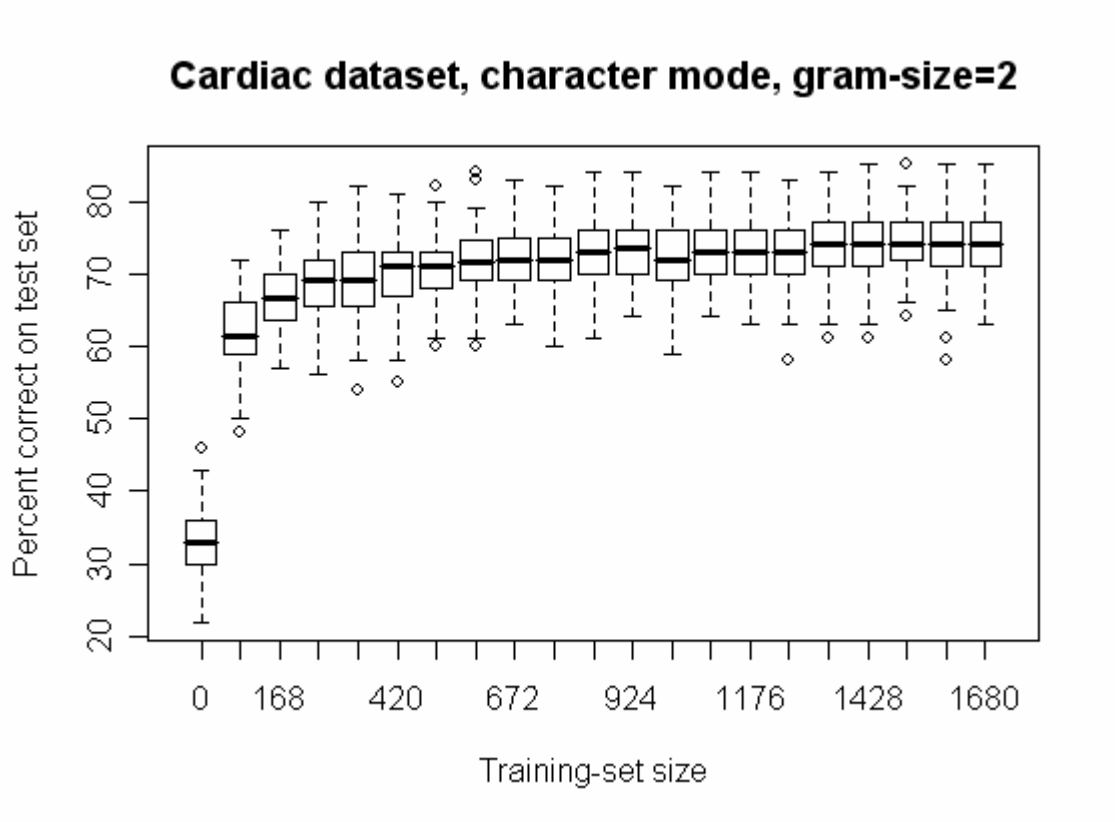

Figure 1. Boxplot illustrating learning system's progress on Cardiac dataset.

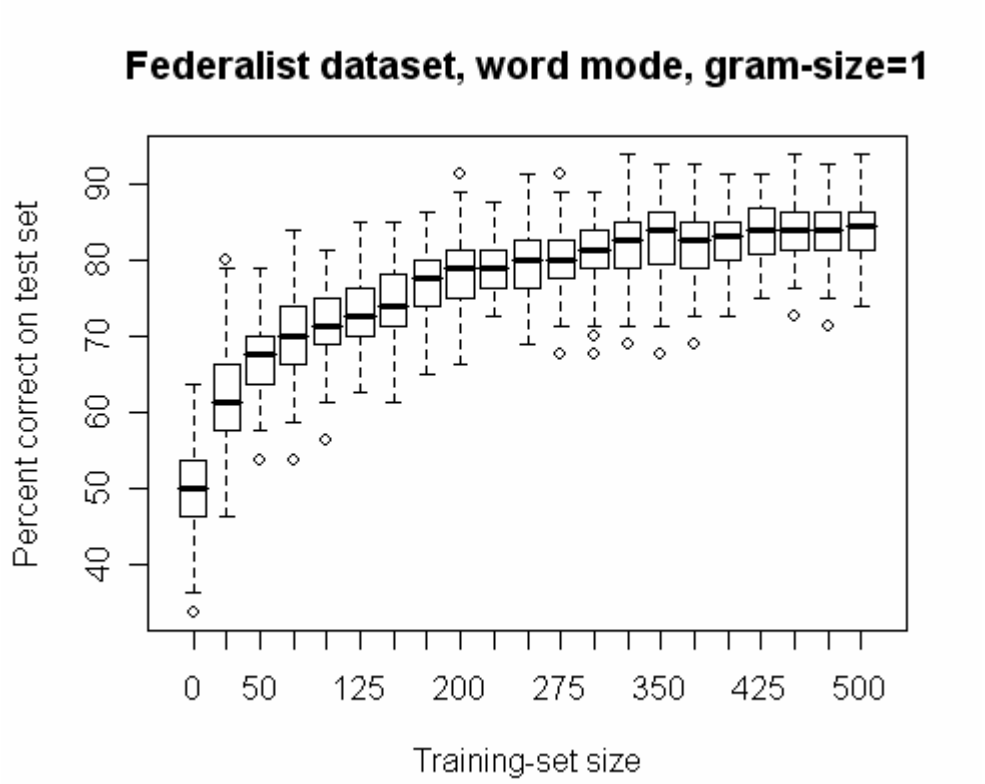

Figure 2. Boxplot illustrating learning system's progress on Federalist Data.

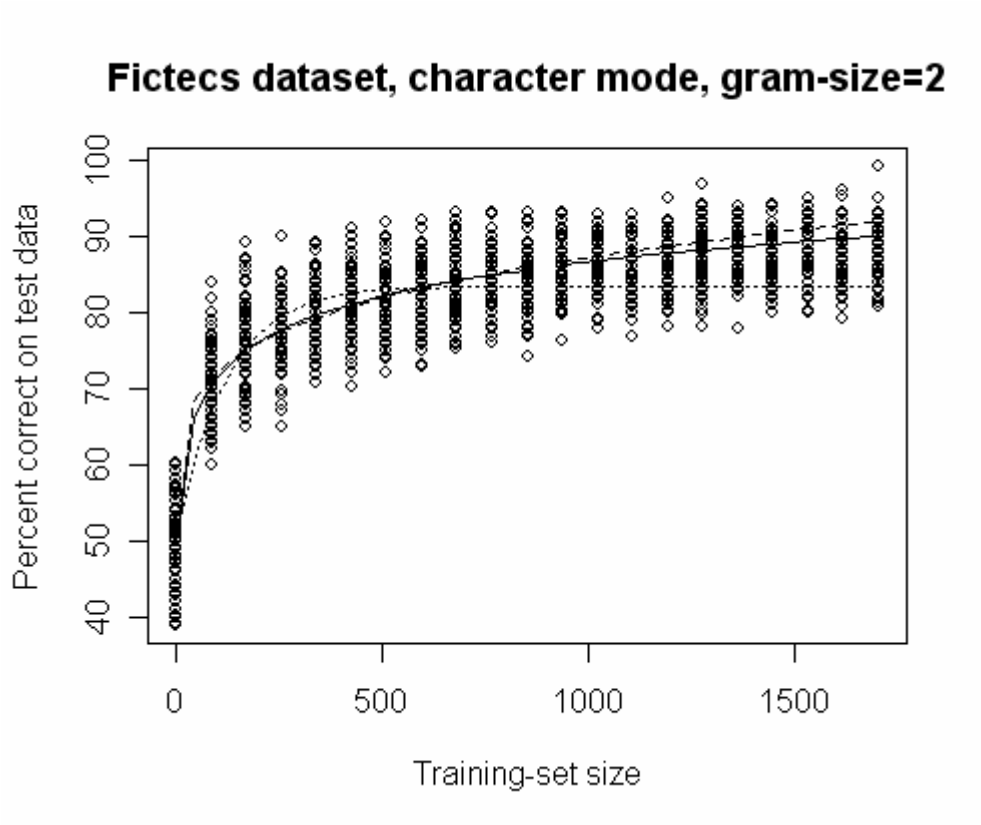

Figure 3. Interpolation and extrapolation on Fictecs data: dotted line = Exponential model; dashed line = Power law; solid line = Log-inverse formula.

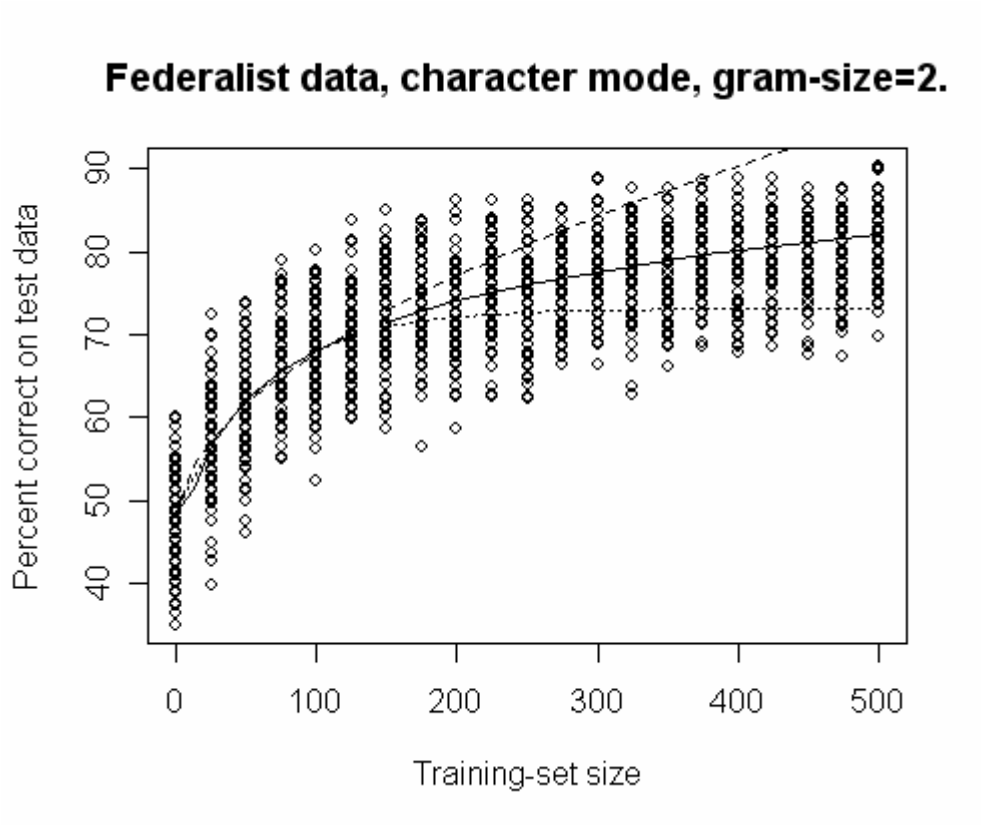

Figure 4. Interpolation and extrapolation on Federalist data: dotted line = Exponential model; dashed line = Power law; solid line = Log-inverse formula.

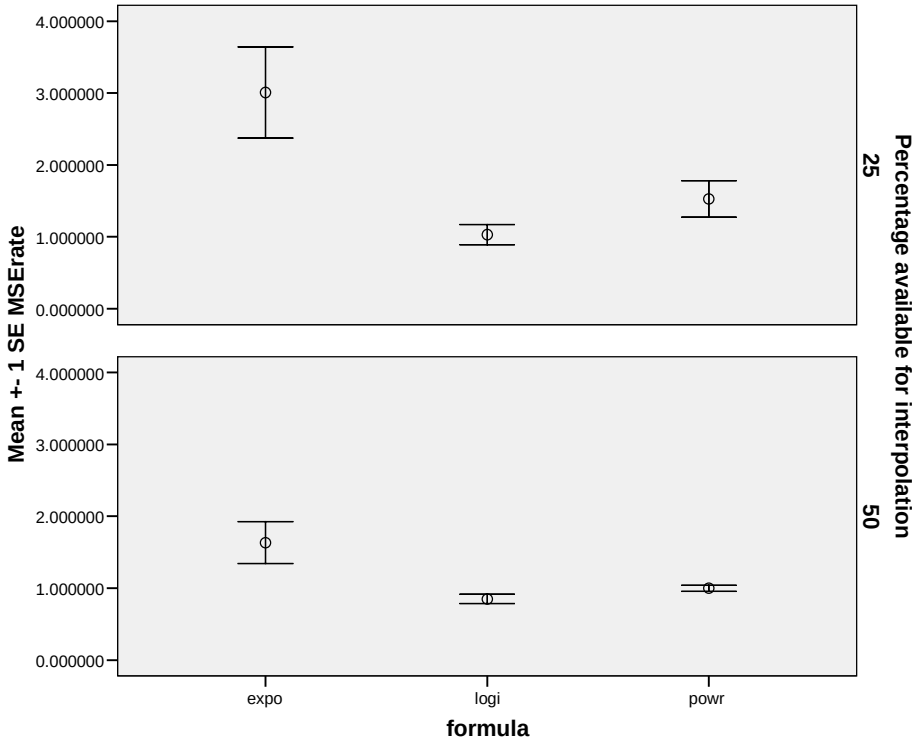

Figure 5. Mean MSE ratios for extrapolations, plus or minus 1 standard error, for all three models in both interpolation conditions.