# Predicting accuracy on large datasets from smaller pilot data

**Mark Johnson**
Macquarie University
Mark.Johnson@MQ.edu.au

**Peter Anderson**
Australian National University
Peter.Anderson@ANU.edu.au

**Mark Dras**
Macquarie University
Mark.Dras@MQ.edu.au

**Mark Steedman**
University of Edinburgh
steedman@inf.ed.ac.uk

## Abstract

Because obtaining training data is often the most difficult part of an NLP or ML project, we develop methods for predicting how much data is required to achieve a desired test accuracy by extrapolating results from systems trained on a small pilot training dataset. We model how accuracy varies as a function of training size on subsets of the pilot data, and use that model to predict how much training data would be required to achieve the desired accuracy. We introduce a new performance extrapolation task to evaluate how well different extrapolations predict system accuracy on larger training sets. We show that details of hyperparameter optimisation and the extrapolation models can have dramatic effects in a document classification task. We believe this is an important first step in developing methods for estimating the resources required to meet specific engineering performance targets.

## 1 Introduction

An engineering discipline should be able to predict the cost of a project before the project is started. Because training data is often the most expensive part of an NLP or ML project, it is important to estimate how much training data required for a system to achieve a target accuracy. Unfortunately our field only offers fairly impractical advice, e.g., that more data increases accuracy (Banko and Brill, 2001); we currently have no practical methods for estimating how much data or what quality of data is required to achieve a target accuracy goal. Imagine if bridge construction was planned the way we build our systems!

Our long-term goal is to develop practical methods for designing systems that achieve target performance specifications, including identifying the amount of training data that the system will require. This paper starts to address this goal by introducing an extrapolation methodology that predicts a system's accuracy on a larger dataset from its performance on subsets of much smaller pilot data. These extrapolations allow us to estimate how much training data a system will require to achieve a target accuracy. We focus on a specific task (document classification) using a specific system (the fastText classifier of Joulin et al. (2016)), and leave to future work to determine if our approach and results generalise to other tasks and systems.

We introduce an *accuracy extrapolation task* that can be used to evaluate different extrapolation models. We describe three well-known extrapolation models and evaluate them on a *document classification dataset*. On our development data the biased power-law method with binomial item weighting performs best, so we propose it should be a baseline for future research. We demonstrate the importance of hyperparameter optimisation on each different-sized data subset (rather than just optimising on the largest data subset) and item weighting, and show that these can have a dramatic impact on extrapolation, especially from small pilot data sets. The data and code for all experiments in this paper, including the R code for the graphics, is available from `http://web.science.mq.edu.au/~mjohnson`.

## 2 Related work

Power analysis (Cohen, 1992) is widely-used statistical technique (e.g., in biomedical trials) for predicting the number of measurements required in an experimental design; we aim to develop sim-

*Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 450–455
Melbourne, Australia, July 15 - 20, 2018.

ilar techniques for NLP and ML systems. There is a large body of research on the relationship between training data size and system performance. Geman et al. (1992) decompose the squared error of a model into a *bias term* (due to model errors) and a *variance term* (due to statistical noise). Bias does not vary with training data size $n$, but the error due to variance should decrease as $O({}^{1}\!/\!\sqrt{n})$ if the training observations are independent (Domingos, 2000a,b). The power-law models used in this paper have been investigated many times in prior literature (Haussler et al., 1996; Mukherjee et al., 2003; Figueroa et al., 2012; Beleites et al., 2013; Hajian-Tilaki, 2014; Cho et al., 2015). Sun et al. (2017), Barone et al. (2017) and the concurrent unpublished work by Hestness et al. (2017) point out that these power-law models describe modern ML and NLP systems quite well, including complex deep-learning systems, so we expect our results to generalise to these systems.

This paper differs from prior work in that we explicitly focus on the task of extrapolating system performance from small pilot data. We introduce a new evaluation task to compare the effectiveness of different models for this extrapolation, and demonstrate the importance of per-subset hyperparameter optimisation and item weighting, which prior work did not investigate.

## 3 Models for extrapolating pilot data

We are given a system whose accuracy on a large dataset we wish to predict, but only a smaller pilot dataset is available. We train the system on different-sized subsets of the pilot dataset, and use the results of those training runs to estimate how the system's accuracy varies as a function of training data size.

We focus on predicting the minimum error rate $e(n)$ that the system can achieve on a dataset of size $n$ after hyperparameter optimisation (where the error rate is $1-$accuracy for a classifier) given a pilot dataset of size $m \ll n$ (in the task below, $m = {}^{n}\!/2$ or $m = {}^{n}\!/10$). We investigate three different extrapolation models of $e(n)$ in this paper:

- *Power law:* $\hat{e}(n) = bn^c$
- *Inverse square-root:* $\hat{e}(n) = a + bn^{-1/2}$
- *Biased power law:* $\hat{e}(n) = a + bn^c$

Here $\hat{e}(n)$ is the estimate of $e(n)$, and $a$, $b$ and $c$ are adjustable parameters that are estimated based on the system's performance on the pilot dataset.

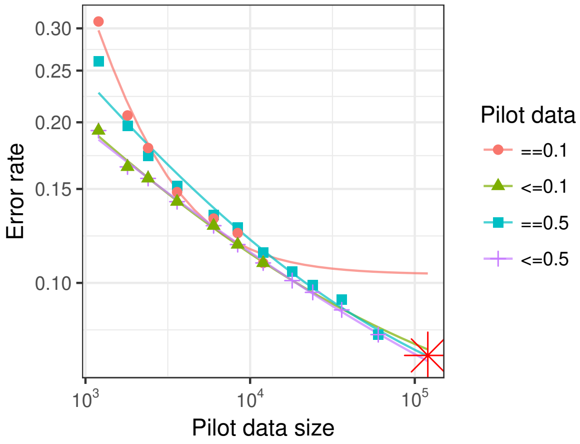

Figure 1: An extrapolation run from pilot data consisting of either 0.1 or 0.5 of the ag_news corpus. The $x$-axis is the size of the subset of pilot data, while the $y$-axis is the classification error rate. The shapes/colors show the maximum fraction of the corpus used in the pilot data, and whether hyperparameters were optimised only once on all of the pilot data (e.g., $= 0.1$ and $= 0.5$) or at each smaller subset of the pilot data (e.g., $\leq 0.1$ and $\leq 0.5$). The lines are least-squares fits of biased power-law models ($\hat{e}(n) = a + bn^c$) to the corresponding pilot data. The red star shows minimum error rate when all the training data is used to train the classifier (this is the value we are trying to predict).

The inverse square-root curve is what one would expect if the error is distributed according to a Bias-Variance decomposition (Geman et al., 1992) with a constant bias term $a$ and a variance term that asymptotically follows the Central Limit Theorem. We fit these models using *weighted least squares regression*. Each data point or *item* in the regression is the result of a run of the system on a subset of the pilot dataset.

Assuming that the underlying system has adjustable hyperparameters, the question arises: how should the hyperparameters be set? The computationally least demanding approach is to optimise the system's hyperparameters on the full pilot dataset, and use these hyperparameters for all the runs on subsets of the pilot dataset. An alternative, computationally more demanding approach is to optimise the system's hyperparameters separately on each of the subsets of the pilot dataset. Figure 1 shows an example where optimising the hyperparameters just on the full pilot dataset is clearly inferior to optimising the hyperparameters on each subset of the pilot dataset. We show below that the more demanding approach of optimising on each subset is superior, especially when extrapolating from small pilot datasets.

We also investigate how details of the regression fit affect the regression accuracy $\hat{e}(n)$. We experimented with several link functions (we used the default Gaussian link here), but found that these had less impact than adjusting the *item weights* in the regression. Runs with smaller training sets presumably have higher variance, and since our goal is to extrapolate to larger datasets, it is reasonable to place more weight on items corresponding to larger datasets. We investigated three item weighting functions in regression:

- constant weights (1),
- linear weights ($n$), and
- binomial weights ($^{n}/_{e(1-e)}$)

Linear weights are motivated by the assumption that the item variance follows the Central Limit Theorem, while the binomial weights are motivated by the assumption that item variance follows a binomial distribution (see the Supplemental Materials for further discussion). As Figure 2 makes clear, linear weights and binomial weights generally produce more accurate extrapolations than constant weights, so we use binomial weights in our evaluation in Table 2.

## 4 A performance extrapolation task

We used the fastText document classifier and the document classification corpora distributed with it; see Joulin et al. (2016) for full details. FastText's speed and evaluation scripts make it easy to do the experiments described below. We fitted our extrapolation models to the fastText document classifier results on the 8 corpora distributed with the fastText classifier. These corpora contain labelled documents for a document classification task, and come randomised and divided into training and test sections. All our results are on these test sections.

The corpora were divided into development and evaluation corpora (each with train and test splits) as shown in table 1. We use the amazon_review_polarity, sogou_news, yahoo_answers and yelp_review_full corpora as our test set (so these are only used in the final evaluation), while the ag_news, dbpedia, amazon_review_full and yelp_review_polarity were used as development corpora. The development and evaluation sets contain document collections of roughly similar sizes and complexities, but no attempt was made to accurately "balance" the development and evaluation corpora.

We trained the fastText classifier on 13 differently-sized prefixes of each training set that are approximately logarithmically spaced over two orders of magnitude (i.e., varying from $^{1}/_{100}$ to all of the training corpus). To explore the effect of hyperparameter tuning on extrapolation, for each prefix of each training set we trained a classifier on each of 1,079 different hyperparameter settings, varying the n-gram length, learning rate, dimensionality of the hidden units and the loss function (the fastText classifier crashed on 17 hyperparameter combinations; we did not investigate why). We re-ran the entire process 8 times on randomly-shuffled versions of each training corpus.

As expected, the minimum error configuration invariably requires the full training data. When extrapolating from subsets of a smaller pilot set (we explored pilot sets consisting of $0.1$ and $0.5$ of the full training data) there are two plausible ways of performing hyperparameter optimisation. Ideally, one would optimise the hyperparameters for each subset of the pilot data considered (we selected the best-performing hyperparameters using grid search). However, if one is not working with computationally efficient algorithms like fastText, one might be tempted to only optimise the hyperparameters once on all the pilot data, and use the hyperparameters optimised on all the pilot data when calculating the error rate on subsets of that pilot data. As figure 2 and table 2 make clear, selecting the optimal hyperparameters for each subset of the pilot data generally produces better extrapolation results. Figure 1 shows how different ways of choosing hyperparameters can affect extrapolation. As that figure shows, hyperparameters optimised on 50% of the training data perform very badly on 1% of the training data. As figure 2 shows, this can lead simpler extrapolation models such as the power-law to dramatically underestimate the error on the full dataset. Interestingly, more complex extrapolation models, such as the extended power-law model, often do much better.

Based on the development corpora results presented in Figures 1 and 2, we choose the biased power law model ($\hat{e}(n) = a + bn^{c}$) with binomial

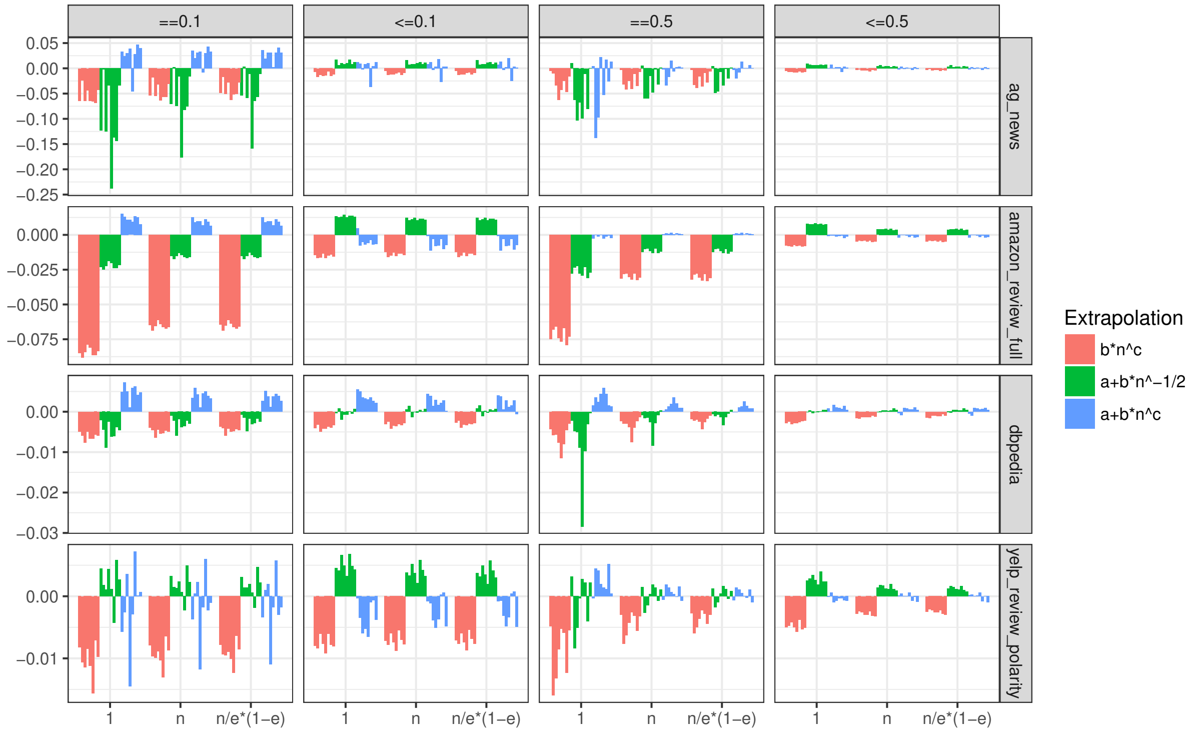

Figure 2: Residuals on 8 runs when extrapolating from pilot data consisting of 0.1 or 0.5 of each development training corpus. The $y$-axis shows the residual error (the difference between the predicted error and the minimum error when the classifier is trained on all the training data), and the $x$-axis indicates the weight function used in extrapolation. Colours indicate the model fitted, (i.e., power-law ($\hat{e}(n) = bn^c$), inverse square-root ($\hat{e}(n) = a + bn^{-1/2}$), or biased power-law ($\hat{e}(n) = a + bn^c$) models). Facets indicate the development corpus used, and whether hyperparameters were optimised only once on all of the pilot data (e.g., $= 0.5$ and $= 0.1$) or on each subset of the pilot data (e.g., $\leq 0.5$ and $\leq 0.1$).

| Corpus | Labels | Train (K) | Test (K) |
|---|---|---|---|
| ag_news | 4 | 120 | 7.6 |
| dbpedia | 14 | 560 | 70 |
| amazon_review_full | 5 | 3,000 | 650 |
| yelp_review_polarity | 2 | 560 | 38 |
| amazon_review_polarity | 2 | 3,600 | 400 |
| sogou_news | 5 | 450 | 60 |
| yahoo_answers | 10 | 1,400 | 60 |
| yelp_review_full | 5 | 650 | 50 |

Table 1: Summary statistics of the development corpora (above line) and evaluation corpora (below line).

| Pilot data | amazon review polarity | sogou news | yahoo answers | yelp review full | **Overall** |
|---|---|---|---|---|---|
| $= 0.1$ | 0.1016 | 0.2752 | 0.0519 | 0.0496 | 0.1510 |
| $\leq 0.1$ | **0.0209** | **0.1900** | **0.0264** | **0.0406** | **0.0986** |
| $= 0.5$ | 0.0338 | 0.0438 | 0.0254 | 0.0160 | 0.0315 |
| $\leq 0.5$ | **0.0049** | **0.0390** | **0.0053** | **0.0046** | **0.0200** |

Table 2: RMS relative residuals ($\hat{e}/e - 1$) on the four evaluation corpora over all runs for the biased power law model ($\hat{e}(n) = a + bn^c$) with binomial item weights ($^n/_{e(1-e)}$). Lower scores are better.

item weights ($^n/_{e(1-e)}$) as the model to evaluate on the evaluation corpora.

We evaluate an extrapolation by calculating the root-mean-square (RMS) of the relative residuals $\hat{e}/e - 1$, where $e$ is the minimum error achieved by the classifier with any hyperparameter setting when trained on the full training set, and $\hat{e}$ is the predicted error made by the extrapolation model from the pilot dataset.[1]

Unsurprisingly, Table 2 shows that extrapolation is more accurate from larger pilot datasets; increasing the size of the pilot dataset 5 times re-

[1] We use relative residuals because the residuals themselves vary greatly from corpus to corpus, and we use RMS to penalise large extrapolation errors. We admit that RMS relative residuals is probably not a close approximation to the extrapolation loss in real applications, and we hope future work will develop more realistic loss functions.

duces the RMS relative residuals by a factor of 10. It also clearly shows that it valuable to perform hyperparameter optimisation on all subsets of the pilot dataset, not just on the whole pilot data. Interestingly, Table 2 shows that the RMS difference between the two approaches to hyperparameter setting is greater when the pilot data is larger. This makes sense; the hyperparameters that are optimal on a large pilot dataset may be far from optimal on a very small subset (this is clearly visible in Figure 1, where the items deviating most are those for the $= 0.5$ pilot data and hyperparameter choice).

## 5 Conclusions and Future Work

This paper introduced an extrapolation methodology for predicting accuracy on large dataset from a small pilot dataset, applied it to a document classification system, and identified the biased power-law model with binomial weights as a good baseline extrapolation model. This only scratches the surface of performance extrapolation tasks. We hope that teams with greater computational resources will study the extrapolation task for computationally more-demanding systems, including popular deep learning models. The power-law models should be considered baselines for more sophisticated extrapolation models, which might exploit more information than just accuracy on subsets of the pilot data.

We hope this work will spur the development of better methods for estimating the resources needed to build an NLP or ML system to meet a specification, as we believe this is essential for any mature engineering field.

## Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This research was supported by a Google award through the Natural Language Understanding Focused Program, and under the Australian Research Council's Discovery Projects funding scheme (project number DP160102156).

## References

M. Banko and Eric Brill. 2001. Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing. In *Human Language Technology Conference : Proceedings of HLT 2001 : First International Conference on Human Language Technology Research*. Morgan Kaufmann.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. *CoRR* abs/1707.09920.

Claudia Beleites, Ute Neugebauer, Thomas Bocklitz, Christoph Krafft, and Jürgen Popp. 2013. Sample size planning for classification models. *Analytica chimica acta* 760:25–33.

J. Cho, K. Lee, E. Shin, G. Choy, and S. Do. 2015. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv:1511.06348* .

Jacob Cohen. 1992. A power primer. *Psychological bulletin* 112(1):155.

Pedro Domingos. 2000a. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning*. pages 231–238.

Pedro Domingos. 2000b. A unified bias-variance decomposition for zero-one and squared loss. *AAAI/IAAI* 2000:564–569.

Rosa L Figueroa, Qing Zeng-Treitler, Sasikiran Kandula, and Long H Ngo. 2012. Predicting sample size required for classification performance. *BMC medical informatics and decision making* 12(1):8.

Stuart Geman, Elie Bienenstock, and René Doursat. 1992. Neural networks and the bias/variance dilemma. *Neural Computation* 4:1–58.

Karimollah Hajian-Tilaki. 2014. Sample size estimation in diagnostic test studies of biomedical informatics. *Journal of biomedical informatics* 48:193–204.

David Haussler, Michael Kearns, H. Sebastian Seung, and Naftali Tishby. 1996. Rigorous learning curve bounds from statistical mechanics. *Machine Learning* 25(2).

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. Deep learning scaling is predictable, empirically. *arXiv:1712.00409* .

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv:1607.01759* .

Sayan Mukherjee, Pablo Tamayo, Simon Rogers, Ryan Rifkin, Anna Engle, Colin Campbell, Todd R Golub, and Jill P Mesirov. 2003. Estimating dataset size requirements for classifying DNA microarray data. *Journal of computational biology* 10(2):119–142.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. *arXiv:1707.02968* .