How much data is enough? Predicting how accuracy varies with training data size

Mark Johnson (with Dat Quoc Nguyen)

Macquarie University
Sydney, Australia

September 4, 2017

Outline

Introduction

Empirical models of accuracy vs training data size

Extrapolating accuracy in NLP applications

Related work

Conclusion

ML as an engineering discipline

- A mature engineering discipline should be able to predict the cost of a project before it starts
- Collecting/producing training data is typically the most expensive part of an ML or NLP project
- We usually have only the vaguest idea of how accuracy is related to training data size and quality
  - More data produces better accuracy
  - Higher quality data (closer domain, less noise) produces better accuracy
  - But we usually have no idea how much data or what quality of data is required to achieve a given performance goal
- Imagine if engineers designed bridges the way we build systems!

Goals of this research project

- Given desiderata (accuracy, speed, computational and data resource pricing, etc.) for an ML/NLP system, design for a system that meets these
- Example: design a classifier that identifies terrorism-related tweets with at least 1% precision and 50% recall and handles 1M tweets/sec. Sample terrorism-related tweets cost \$1 each, while random tweets cost $\$10^{-5}$ each.
  - What hardware/software should I use?
  - *How many of each kind of tweet should I buy?*

What this paper contributes

- Studies how accuracy varies as a function of training data size for several NLP models and tasks
- Discusses three methods for extrapolating accuracy predictions as a function of training data size
- Proposes a new *accuracy extrapolation* task, provides datasets and results for the three extrapolation methods

Outline

Introduction

Empirical models of accuracy vs training data size

Extrapolating accuracy in NLP applications

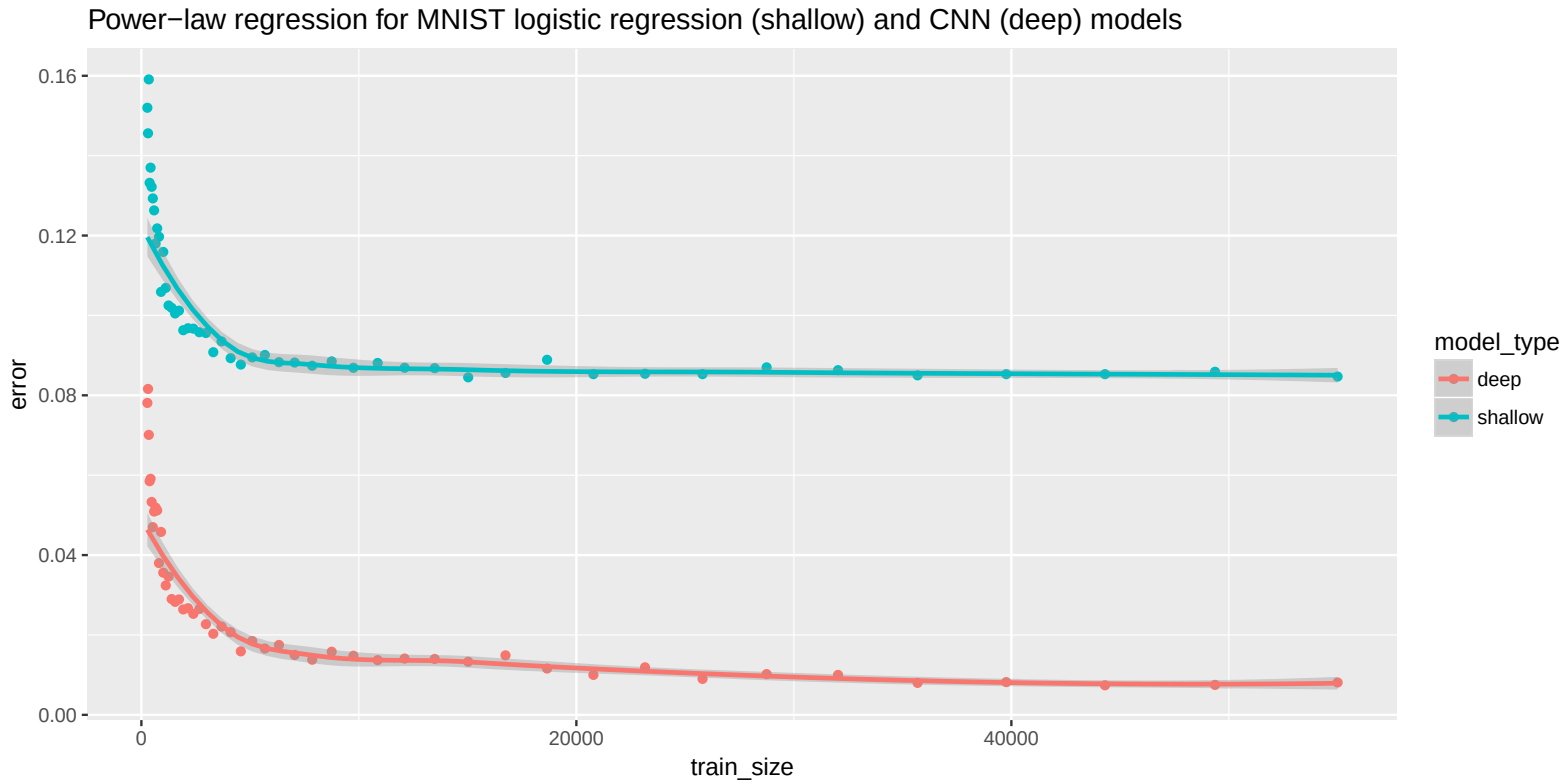
Related work

Conclusion

Overview

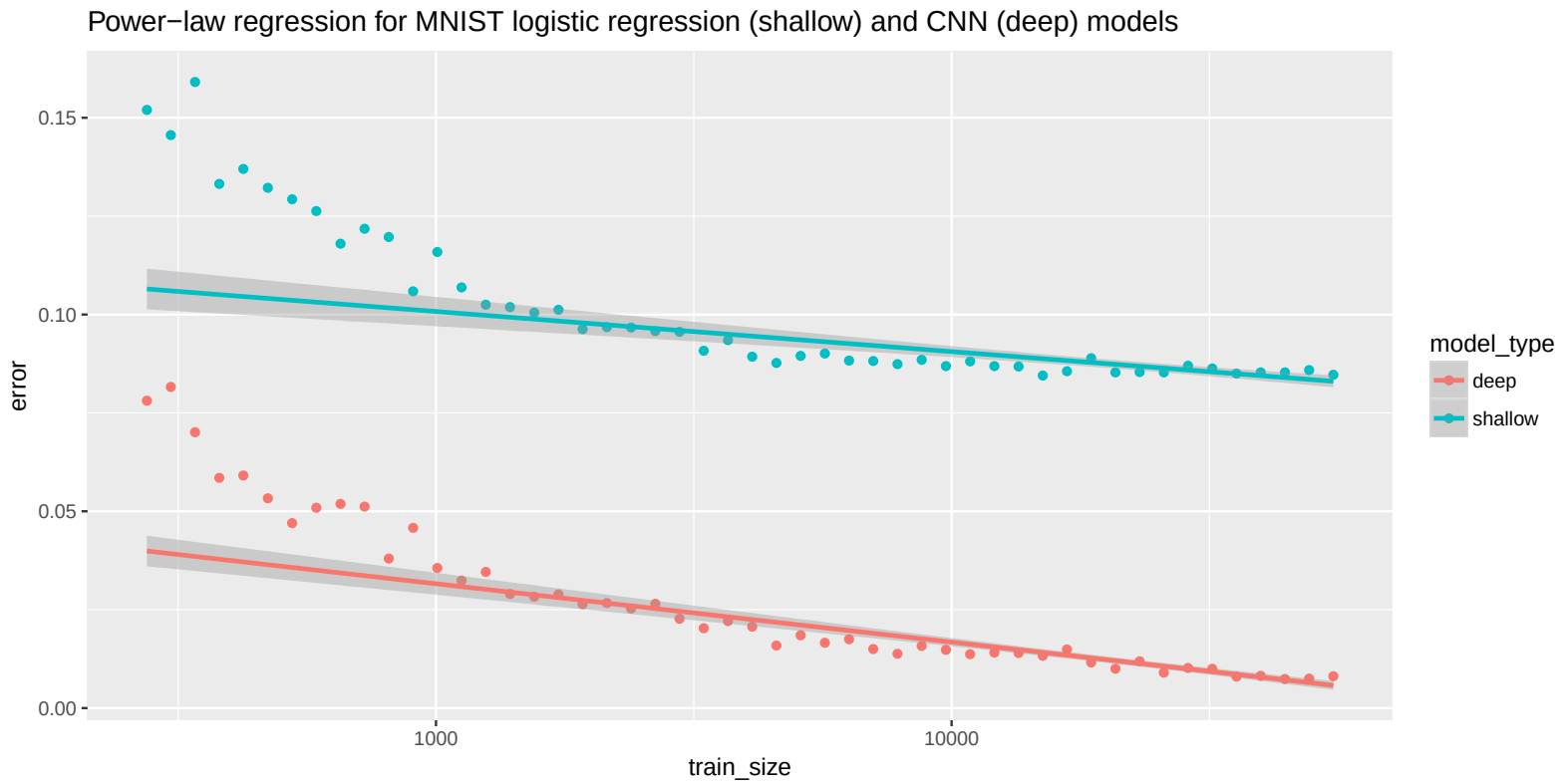
- Three models of how Error (= 1 – *accuracy*) depends on training data size n
  - *Power law:* Error $= b\, n^{-c}$
  - *Inverse square root:* Error $= a + b\, n^{-1/2}$
  - *Extended power law:* Error $= a + b\, n^{-c}$
- Parameters estimated from multiple runs using *weighted least squares regression*
  - Model is run on different-sized subsets of training data
  - Same test set is used to evaluate each run
  - The evaluation of each model training/test run is a data point
  - Each data point (run) is weighted by training data size n
  - Perhaps another loss function would be more motivated?
  - If evaluation returns f-score, assume Error = 1 – f-score?

Error vs training size: MNIST digits (1)



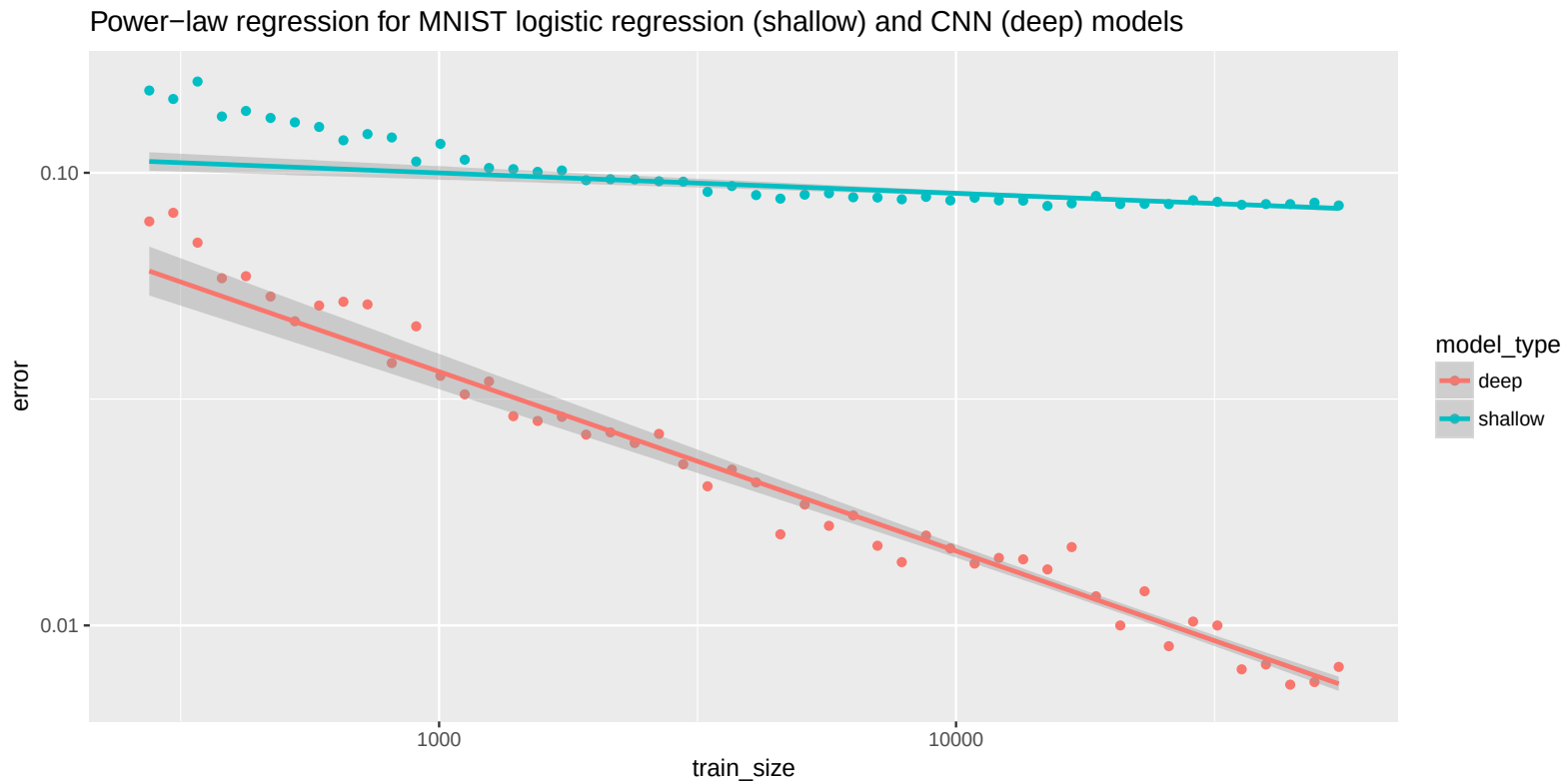
- Error = 1 – Accuracy
- Error and training size axes have linear scale
  - Highly non-linear relationship
  - Non-linear regression (loess) to fit error curve

Error vs training size: MNIST digits (2)



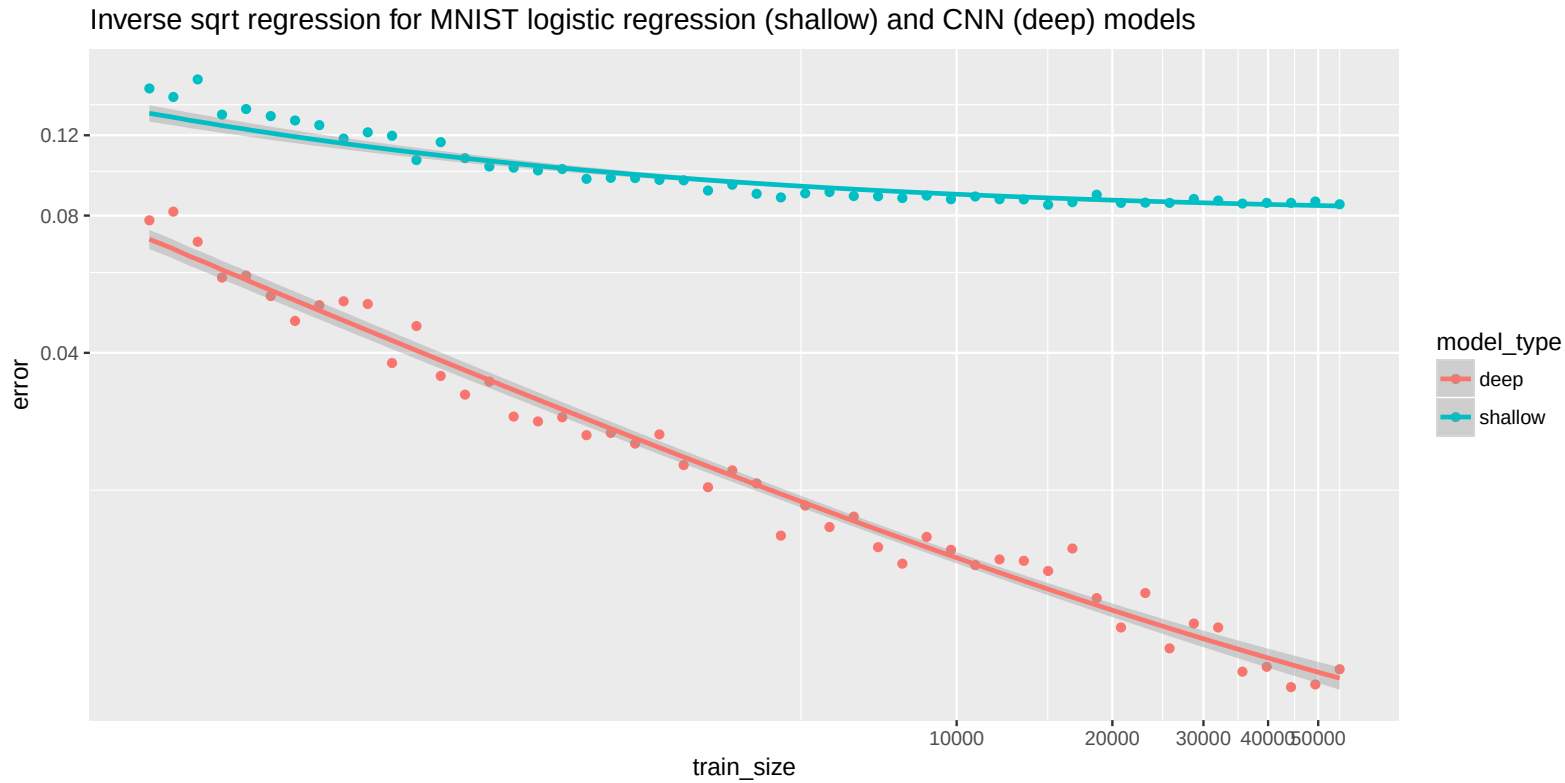
- Error = 1 – Accuracy
- Error axis has linear scale, training size axis has log scale
  - Linear regression to fit error curve

Power-law relationship



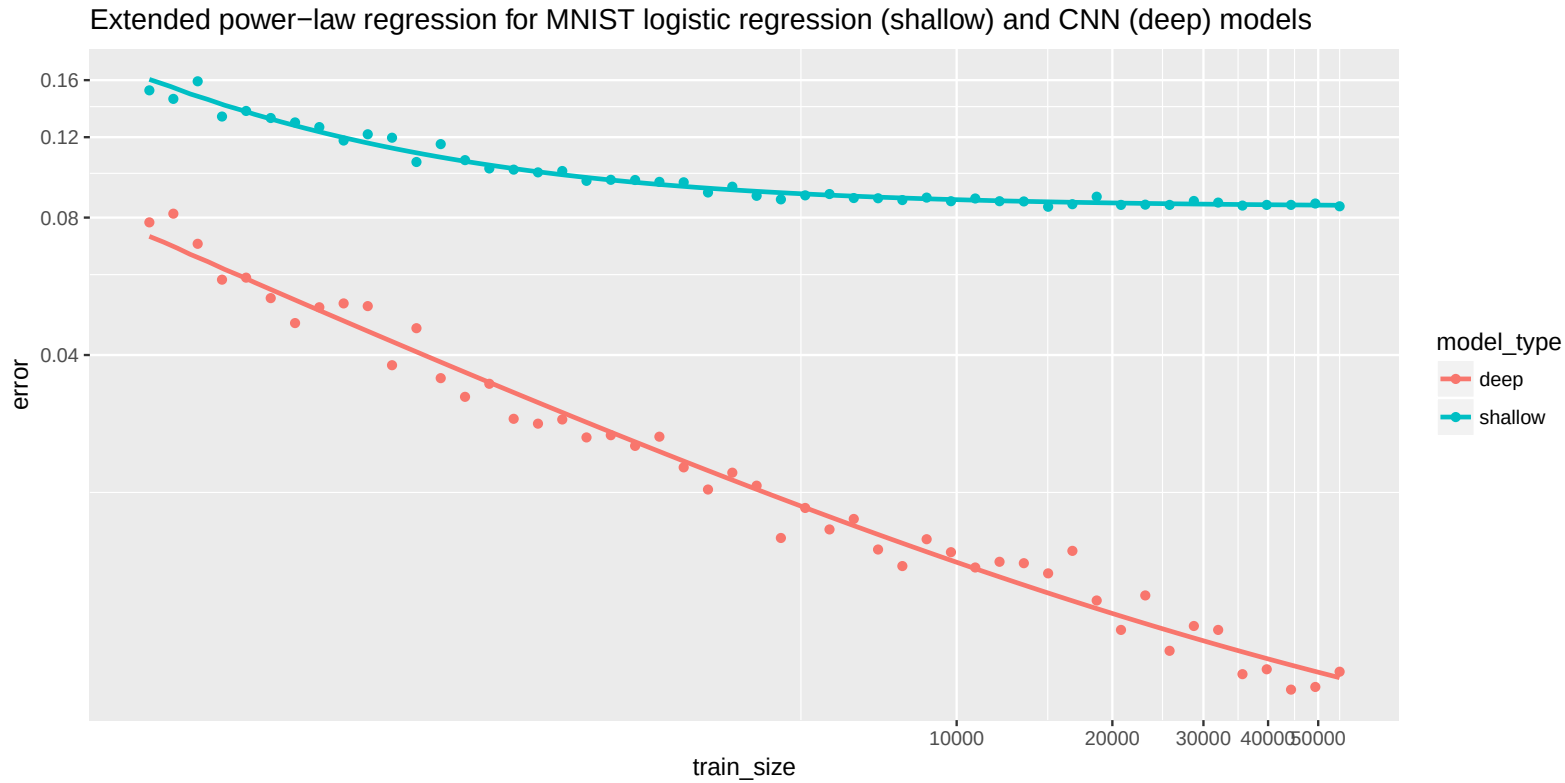
- Error $= b\, n^{-c}$, where n = training data size
- Predicts that Error $\to 0$ as $n \to \infty$ if $c > 0$
- Linear relationship between $\log(\text{Error})$ and $\log(n)$

Inverse square-root relationship



- Error $= a + b\, n^{-1/2}$, where n = training data size
- Predicts that Error $\longrightarrow a$ as $n \longrightarrow \infty$
- Inspired by *Bias-Variance decomposition* (Geman et al., 1992)
  - a is a *bias term* due to model mis-specification
  - From Central Limit Theorem, variance $\propto 1/n$

Extended power law relationship



- Error $= a + b\,n^{-c}$, where n = training data size
- Predicts that Error $\longrightarrow a$ as $n \longrightarrow \infty$ if $c > 0$
- $c = 1/2$ (inverse sqrt) assumes test items are *independent*

$\Rightarrow$ $c < 1/2$ if there are dependencies among test items

- Estimating these parameters involves non-linear least-squares optimisation, which can be unstable or fail

Using an accuracy model to predict data requirements

- High-level description:
  - Determine error rate of target system on data sets of various sizes
  - Estimate parameters of accuracy model
  - Find the training size $\hat{n}$ that the accuracy model predicts achieves the desired error rate
- More sophisticated approaches:
  - Use *bootstrap resampling* for confidence intervals on $\hat{n}$

Outline

Introduction

Empirical models of accuracy vs training data size

Extrapolating accuracy in NLP applications

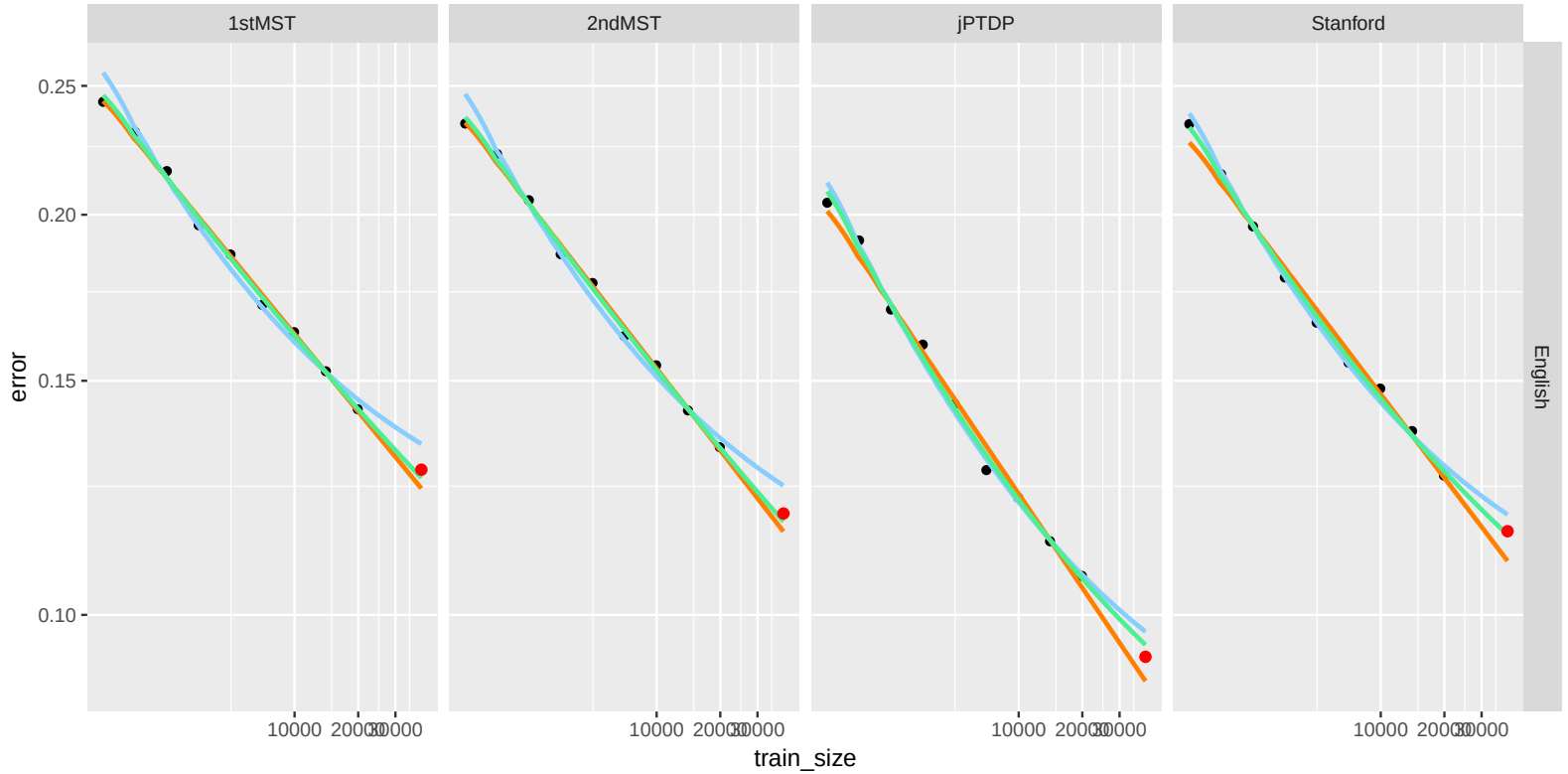
Related work

Conclusion

Error extrapolation task

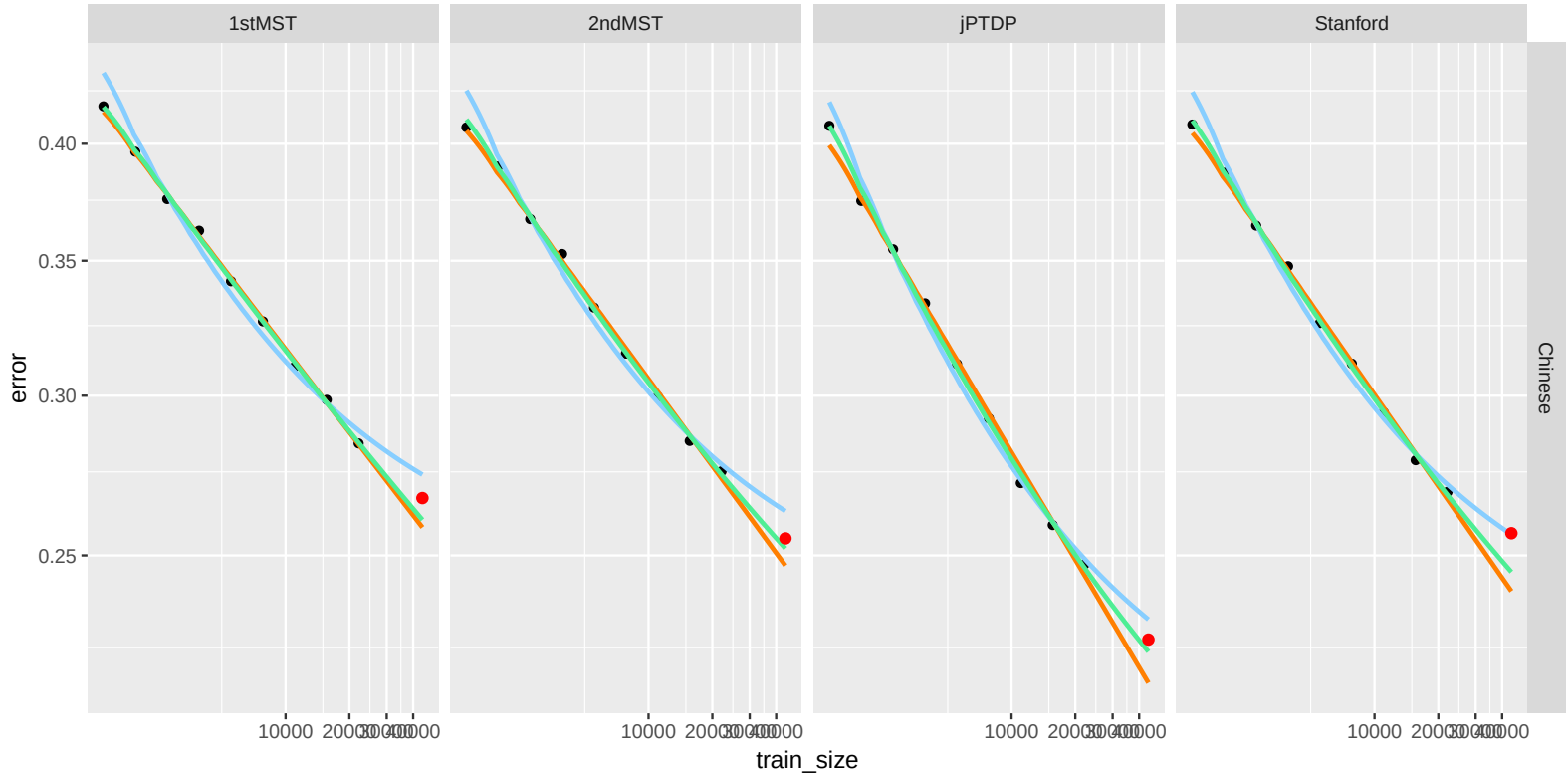
- Given error on training data sets of size n/k (where $k = 2$ or 10) or less, predict error on data set size n.
  - Report *absolute difference* of predicted and true error
  - Perhaps an asymmetric loss would be more appropriate?
- All evaluations use same test set
- The training data subsets are all contained in the same subset of size n/k
  - Motivation: the only training data you have is of size n/k, but you can do anything you want with it

Extrapolating English dependency parsing



- Black points: training error
- Red point: test error (which we are predicting)
- Orange: power law relationship, Error $= b\,n^{-c}$
- Blue: inverse sqrt relationship, Error $= a + b\,n^{-1/2}$
- Green: extended power law relationship, Error $= a + b\,n^{-c}$

Extrapolating Chinese dependency parsing



- Black points: training error
- Red point: test error (which we are predicting)
- Orange: power law relationship, Error $= b\,n^{-c}$
- Blue: inverse sqrt relationship, Error $= a + b\,n^{-1/2}$
- Green: extended power law relationship, Error $= a + b\,n^{-c}$

Dependency parsing, extrapolating ½ data

	language	parser	obs	plaw	isqrt	ext.plaw
1	Chinese	1stMST	9	0.00880	0.00724	0.00656
2	Chinese	2ndMST	9	0.00780	0.00806	0.00293
3	Chinese	jPTDP	9	0.01096	0.00527	0.00313
4	Chinese	Stanford	9	0.01641	0.00037	0.01109
5	English	1stMST	9	0.00412	0.00586	0.00183
6	English	2ndMST	9	0.00367	0.00591	0.00166
7	English	jPTDP	9	0.00383	0.00413	0.00194
8	English	Stanford	9	0.00581	0.00337	0.00067

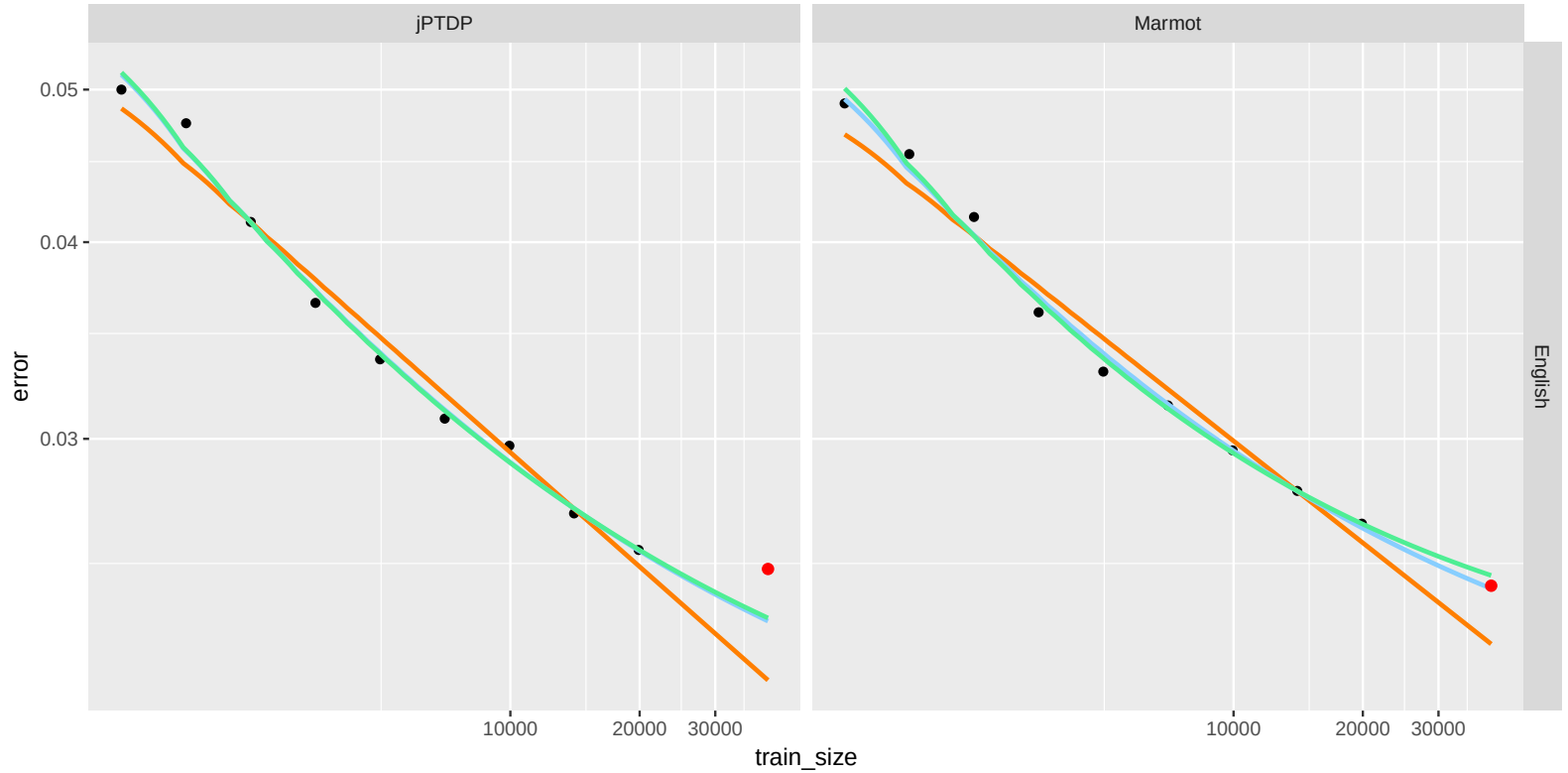
- Extended power law is more accurate than other extrapolations, except for Stanford parser on Chinese

Dependency parsing, extrapolating 1/10 data

	language	parser	obs	plaw	isqrt	ext.plaw
1	Chinese	1stMST	4	0.00760	0.03715	0.04847
2	Chinese	2ndMST	4	0.00545	0.03927	0.02431
3	Chinese	jPTDP	4	0.01665	0.03104	0.05008
4	Chinese	Stanford	4	0.01891	0.02738	0.01873
5	English	1stMST	4	0.00939	0.01998	
6	English	2ndMST	4	0.00973	0.01837	
7	English	jPTDP	4	0.00574	0.01792	0.01098
8	English	Stanford	4	0.01920	0.00741	0.02195

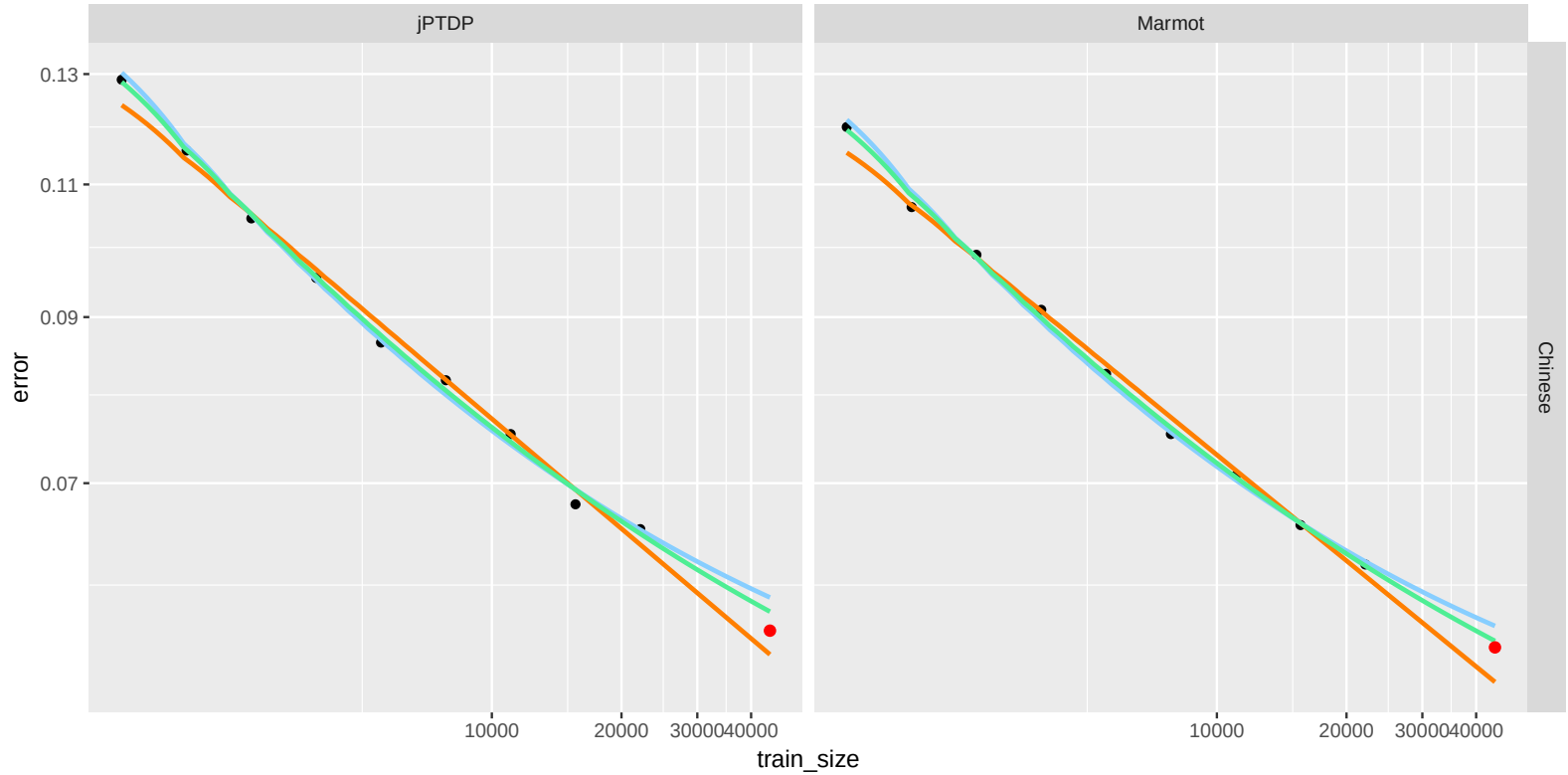
- Extended power law regression failed to converge on 2 settings
- Power law regression gives most accurate extrapolation on 6 settings

Extrapolating English POS tagging



- Black points: training error
- Red point: test error (which we are predicting)
- Orange: power law relationship, Error $= b\,n^{-c}$
- Blue: inverse sqrt relationship, Error $= a + b\,n^{-1/2}$
- Green: extended power law relationship, Error $= a + b\,n^{-c}$

Extrapolating Chinese POS tagging



- Black points: training error
- Red point: test error (which we are predicting)
- Orange: power law relationship, $\text{Error} = b\,n^{-c}$
- Blue: inverse sqrt relationship, $\text{Error} = a + b\,n^{-1/2}$
- Green: extended power law relationship, $\text{Error} = a + b\,n^{-c}$

POS tagging, extrapolating ½ data

	language	tagger	obs	plaw	isqrt	ext.plaw
1	Chinese	jPTDP	9	0.00198	0.00289	0.00164
2	Chinese	Marmot	9	0.00278	0.00180	0.00053
3	English	jPTDP	9	0.00372	0.00182	0.00172
4	English	Marmot	9	0.00198	0.00010	0.00037

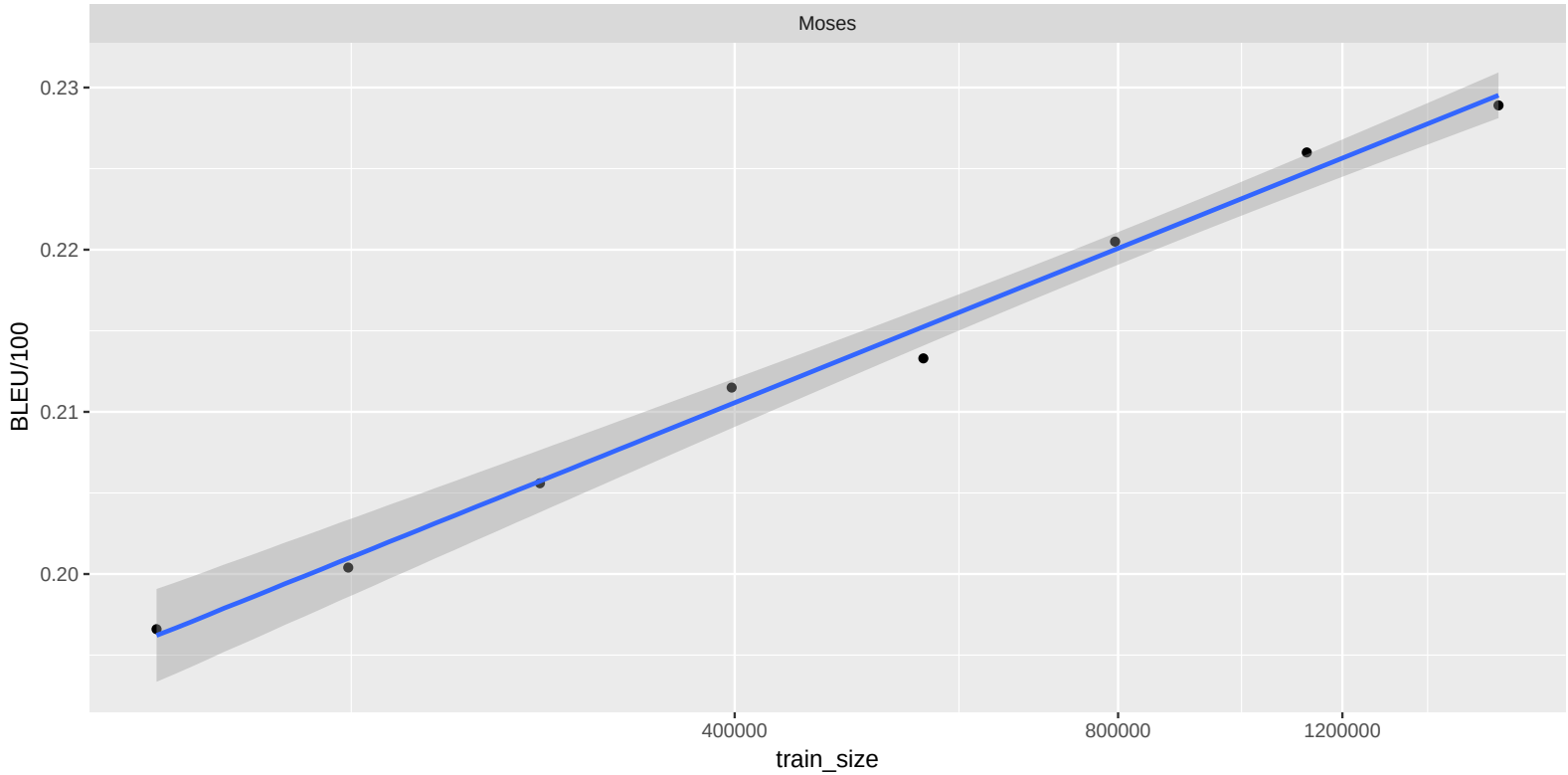
- Extended power law gives most accurate extrapolation on 3 settings

POS tagging, extrapolating 1/10 data

	language	tagger	obs	plaw	isqrt	ext.plaw
1	Chinese	jPTDP	4	0.00867	0.00496	0.00703
2	Chinese	Marmot	4	0.00603	0.00740	0.01932
3	English	jPTDP	4	0.00769	0.00278	
4	English	Marmot	4	0.00634	0.00121	

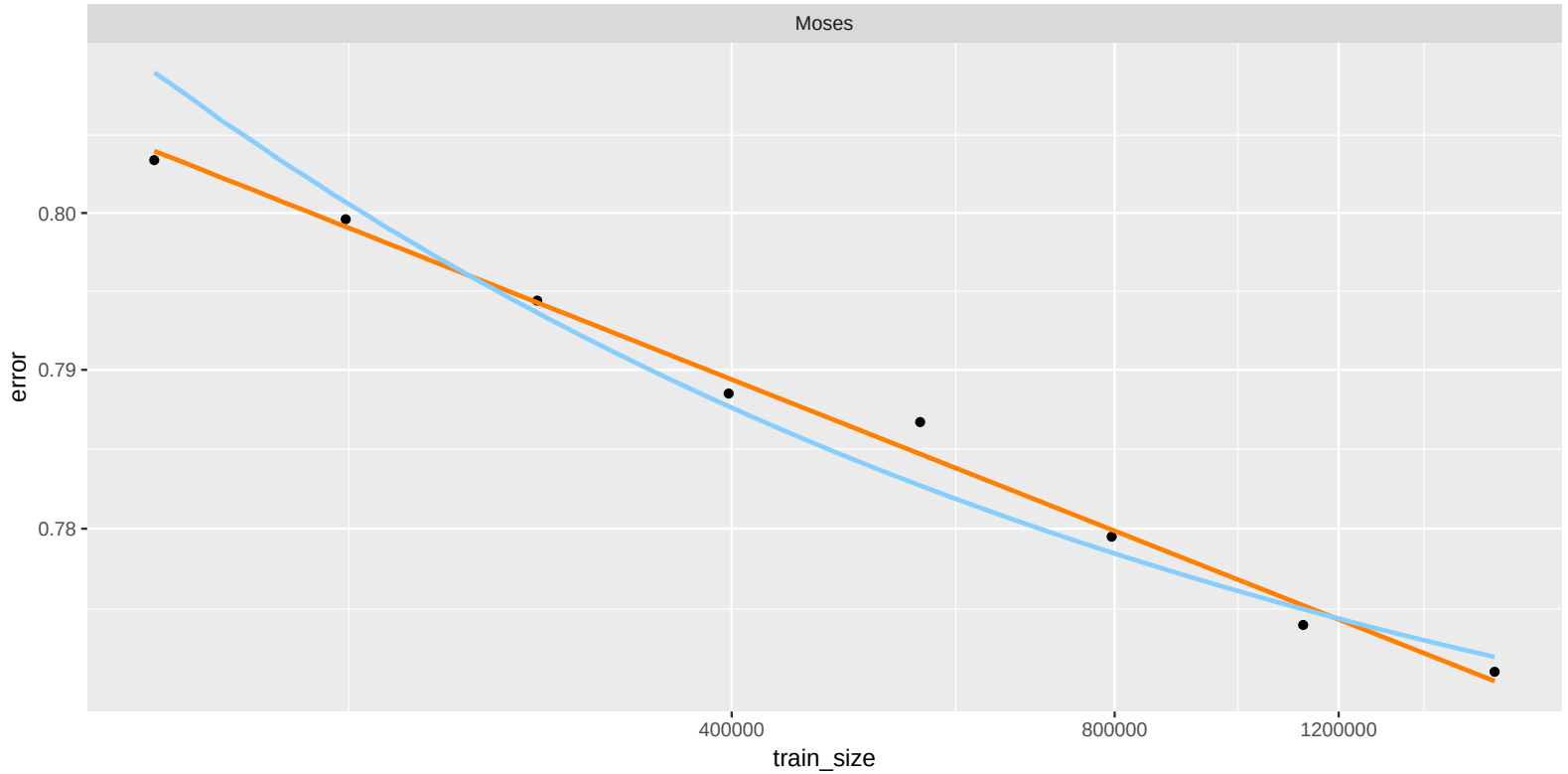
- Extended power law regression failed to converge on 2 settings
- Inverse sqrt regression gives most accurate extrapolation on 3 settings

Machine translation



- BLEU is close to linearly related to log training size
- Predicts that BLEU will grow unboundedly as training data gets larger

Our models applied to Machine Translation



- Black points: training error, where Error $= 1 - \text{BLEU}/100$
- Red point: test error (which we are predicting)
- Orange: power law relationship, Error $= b\,n^{-c}$
- Blue: inverse sqrt relationship, Error $= a + b\,n^{-1/2}$
- Green: extended power law relationship, Error $= a + b\,n^{-c}$ (FAILED TO CONVERGE)

Outline

Introduction

Empirical models of accuracy vs training data size

Extrapolating accuracy in NLP applications

Related work

Conclusion

Predicting accuracy as a function of training size

- Mukherjee et al. (2003) and Figueroa et al. (2012) predict classifier accuracy in a biomedical setting by fitting a power-law curve equivalent to one used here
- Beleites et al. (2013) discuss classifier accuracy with very small training sets (tens of examples) in chemical applications
- Hajian-Tilaki (2014) discusses how ROC and AUC vary with sample size in biomedical applications
- Cho et al. (2015) investigate how much data is needed to train a medical image deep learning system
- Sun et al. (2017) observe that performance of a deep learning machine translation system increases even with very large training data sets

Sample complexity

- *Sample complexity* is the name used in machine learning for the relationship between classifier accuracy and training data size
- Plays an important theoretical role in Empirical Risk Minimisation and Support Vector Machines
- Not studied empirically, AFAIK

Power calculations

- In statistics, a *power calculation* is used to determine how many samples are required in an experiment to test a hypothesis
  - Widely used in drug trials
- Given a hypothesis test and an *effect size* (difference between two conditions), a power calculation returns the sample size for which it is likely that the test will reject the null hypothesis

Bias-Variance Trade-off

- Geman et al. (1992) decompose the squared error of a
- regression model into two terms:
  - A *bias term*, due to model errors
  - A *variance term*, due to statistical noise
- As the model gets more complex, bias decreases but variance increases
- Bias does not vary with training data size n, but variance should decrease as $1/n$ if observations are independent
  - If observations are not independent, variance will decrease more slowly
- Domingos (2000a) and Domingos (2000b) generalise the Bias-Variance decomposition to $0-1$ loss and squared loss
  - They also propose a bootstrap procedure to estimate Bias and Variance

Outline

Introduction

Empirical models of accuracy vs training data size

Extrapolating accuracy in NLP applications

Related work

Conclusion

Conclusion and future work

- If ML and NLP are to become reliable engineering disciplines, we need to be able to predict how much effort a project will require
- Training data is often the most expensive and difficult resource to acquire $\Rightarrow$ need to predict training data requirements
- This paper describes three different procedures for extrapolating the performance of a system on a large training data set from the performance on a smaller data set
- We introduce an extrapolation task that compares extrapolation procedures
- Undoubtedly there are much better ways of extrapolating system performance!

References

Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C., and Popp, J. (2013). Sample size planning for classification models. *Analytica chimica acta*, 760:25–33.

Cho, J., Lee, K., Shin, E., Choy, G., and Do, S. (2015). How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *ArXiv e-prints*.

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

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

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

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58.

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

Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, R., Engle, A., Campbell, C., Golub, T. R., and Mesirov, J. P. (2003). Estimating dataset size requirements for classifying DNA microarray data. *Journal of computational biology*, 10(2):119–142.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. *arXiv preprint arXiv:1707.02968*.