Data

Andrew Shi

January 29, 2025

1 Frameworks for evaluating data supply

Villalobos et al. [11] states that we will run out of public human text data between 2026-2032. To study this problem, we look at two competing resources: the amount of the data available and the rate of data consumption. In this section, we first list the frameworks proposed by [11] to model data supply. Then, we summarize a compute-based model to project data consumption by training language models.

1.1 Future projection of data supply

First, we consider the growth of the indexed web; it is important to note that not all internet data can/should be used to train models, as bad data degrades model output quality. However, there are data cleaning techniques that can be used to salvage a larger proportion of internet data, which we study in Section 1.2. Also note that dataset sizes are calculated based on the number of tokens present. Finally, note that synthetic data is not included in either of these models; we discuss synthetic data in Section 2.

[11] uses the CommonCrawl dataset [2] as a proxy for the size of the indexed web. They estimate there are roughly 510T tokens on the indexed web in 2024. Furthermore, they introduce the functional form

$$S_{IW}(y) = N_{IW} \cdot B_P \cdot T_B \cdot (1+g)^{y-y_0}$$

where $S_{IW}(y)$ is the estimate of the current stock of tokens in the indexed web in a given year y, N_{IW} is the number of unique web pages in the indexed web, B_P is the average number of bytes per web page, T_B is the average number of tokens per byte, and g is the estimated rate of growth of the total number of tokens.

Another approach to forecasting future data supply uses the growth in the number of internet users instead of the number of pages on the indexed web. The authors consider two factors: (1) increases in the human population, and (2) increases in "internet penetration," i.e. the percentage of the population that uses the internet [11]. They fit a sigmoid function to model internet user growth, and ultimately achieve a final estimate of 3100T tokens on the indexed web in the limit [95%: 1900T, 5200T]. This serves as an estimate for the upper bound of tokens on the indexed web.

A third approach is to estimate the historical dataset growth rate and the projected compute growth rate. [11] reports that the size of dataset sizes cannot grow indefinitely, even if there were an infinite data supply due to compute bounds. However, in recent years, computing power has grown significantly, leading to increased demand for larger datasets. Thus, it is logical to forecast dataset size using the historical growth rate and a projection of compute growth rate. The researchers first extrapolate a functional form for the historical dataset size growth:

$$D_H(y) = G_D^{y-y_0} \cdot D(y_0)$$

where D_H is the training dataset size, G_D is the factor growth per year, Y_0 is some base year, and Y is the year. Both G_D and $D_(y0)$ are lognormal distributions.

The compute-based dataset size growth takes the functional form:

$$D_C(y) = \sqrt{\frac{20}{6} \cdot C(y)}$$

where $D_C(y)$ is the projected amount of data used in notable training runs and $C(y)$ is the probabilistic projection of largest compute spent on a training run, modeled following [1]. 6 is the number of FLOP per parameter per token and 20 is the approximate number of training tokens per parameter according to [5].

Finally, the mixture model projection for dataset growth, using equal weights, takes the functional form:

$$F_{D(y)} = \frac{1}{2}\left(F_{D_H(y)} + F_{D_C(y)}\right)$$

1.2 What percent of data on the indexed web can be used to train models?

It is easy to understand that not all public data scraped from the internet should be used to train models, as poor quality data may lead to degradation of model performance. In this section, we discuss a few techniques used to predictably clean data that can be used to effectively train models.

Data filtering [3] and data deduplication [7] are common techniques used to clean raw public data. [9] shows that after applying these two techniques, the dataset size decreases by around 50%. Further pruning of the filtered and deduplicated data, which leads to optimal model performance using a perplexity measurement, results in another 50% reduction of the dataset size. [11] states that their 95% confidence interval for usable internet data [10%, 40%].

1.3 Future projection of data consumption

Combining the models listed in Section 1.1, [11] states that the median exhaustion date for publicly available data is 2028, and becomes highly likely by 2032. They estimate that training models will cost around 5e28 FLOPs when publicly data is fully utilized. If overtraining models becomes common practice, the data bottleneck could happen a year earlier, at a training compute cost of 6e27 FLOPs.

2 Synthetic Data

While public data is limited, there has been a focus on using AI models themselves to generate synthetic data, which in turn is oftentimes used to train new models. While this method has proven to be successful in multiple domains, there are risks for model performance degradation.

2.1 Methods

On the other hand, training on synthetic data has shown much promise in domains where model outputs are relatively easy to verify, such as mathematics, programming, and games. [8] leverages a series of operations to increase the complexity of math questions and answers using GPT-3.5, while [12] bootstraps the questions in MATH and GSM8K by rewriting them in different ways, such as semantic rephrasing, self-verification, and backward reasoning. Both show improved model performance when training on the augmented datasets. For code generation, [4] propose a self-improvement strategy where the models generate their own synthetic puzzle-solution pairs and [10] propose a framework that lever ages a simulated environment and adaptation strategies like self-improvement synthetic data generation and CoT prompting for code optimization. Another domain is a synthetically-generated environment that may be useful for action-reward models commonly used in computer vision, robotics, and agent planning. For instance, [6] leverages natural language form feedback generated by the simulated environment to teach LLM-based robots planning.

2.2 Risks

There are also risks associated with using synthetic data to train models. A model is simply a probability distribution over a vocabulary of tokens; thus, using a pre-trained model to generate synthetic data creates samples that are concentrated around the mean. Thus, using this data to train a new model can potentially lead to mean-field sharpening of the new distribution. This may result in losing the tail-ends of the original distribution, causing the model to be less expressive and limited in its outputs.

3 To Explore

- Can we observe gains in model quality without having to scale data? What if we take this scaling law out of the equation? For example, project growth in compute, take dataset size out of the equation, and maximize based on the other parameters
- Estimates of data depletion categorized by modality
- Are new architectures being developed to be more data efficient?

References

[1] T. Besiroglu, L. Heim, and J. Sevilla. Projecting compute trends in machine learning, 2022. Accessed: 2025-01-23.

[2] C. Crawl. Common crawl. http://commoncrawl.org, 2025.

[3] L. Gao. An empirical exploration in quality filtering of text data, 2021.

[4] P. Haluptzok, M. Bowers, and A. T. Kalai. Language models can teach themselves to program better, 2023.

[5] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models, 2022.

[6] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, N. Brown, T. Jackson, L. Luu, S. Levine, K. Hausman, and B. Ichter. Inner monologue: Embodied reasoning through planning with language models, 2022.

[7] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini. Deduplicating training data makes language models better, 2022.

[8] H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, Y. Tang, and D. Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct, 2025.

[9] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.

[10] A. Shypula, A. Madaan, Y. Zeng, U. Alon, J. Gardner, M. Hashemi, G. Neubig, P. Ranganathan, O. Bastani, and A. Yazdanbakhsh. Learning performance-improving code edits, 2024.

[11] P. Villalobos, A. Ho, J. Sevilla, T. Besiroglu, L. Heim, and M. Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data, 2024.

[12] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024.