

# Opening the Black Box(es)

or, what I wish I had done ten years ago

---

Charles H. Pence •  [@pence@scholar.social](mailto:@pence@scholar.social)

Data Driven Philosophy, 2025-10-17



# A Strange Talk

This is a weird talk: I've found myself getting progressively more worried about the state of DH, and I'm using you all as free therapy.



**MIDLIFE  
CRISIS  
AHEAD**

# Opening the Black Box

---

## Topics

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
comp	Blei (2012)
...	

## Documents

### Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,<sup>1</sup> two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains

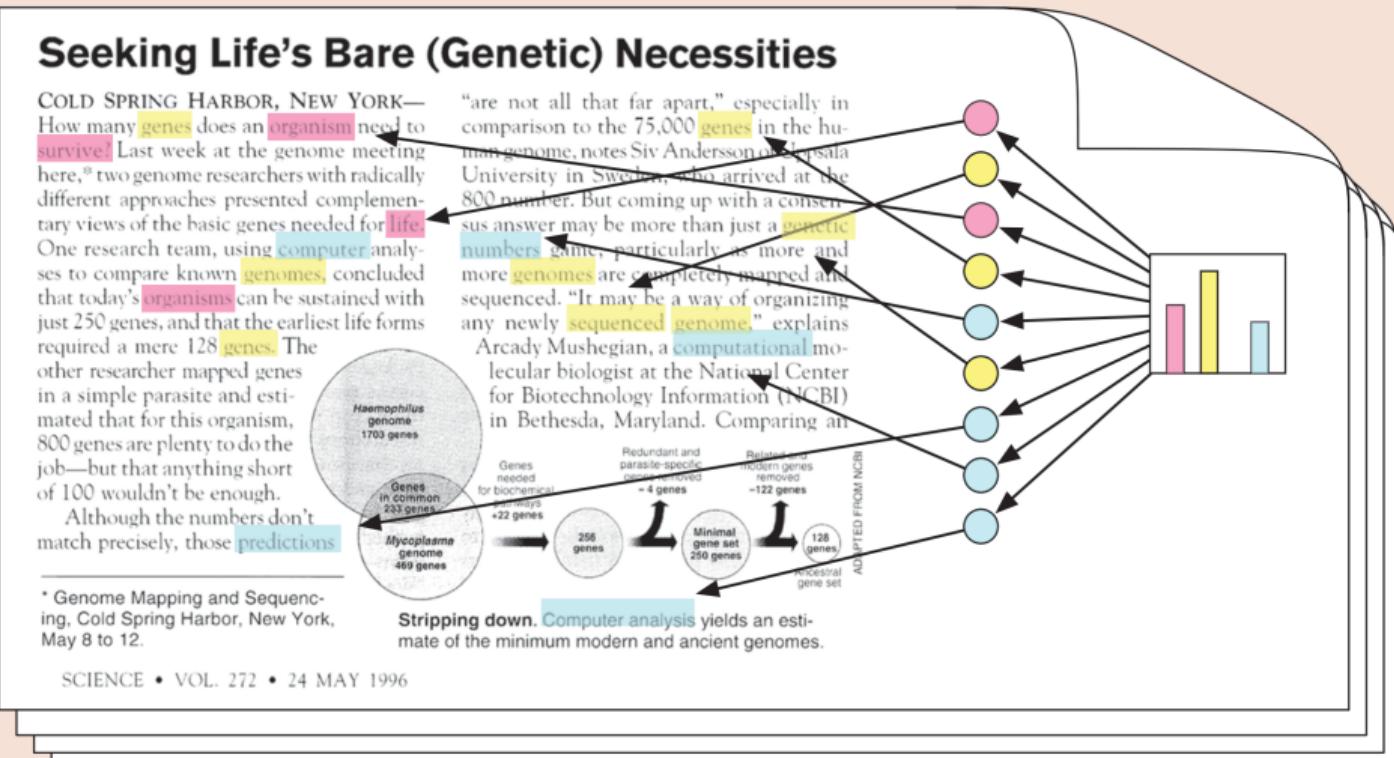
Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

\* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

## Topic proportions and assignments



# A “Gray” Box

- Inference details remain fairly opaque, and can sometimes be very sensitive to user choices (of number of topics, of hyperparameters)
- But! At least you can carefully read **a lot** of intermediates:
  - top words in topics
  - top documents for topics
  - topic probability distributions for selected documents
  - word probability distributions across topics (!)

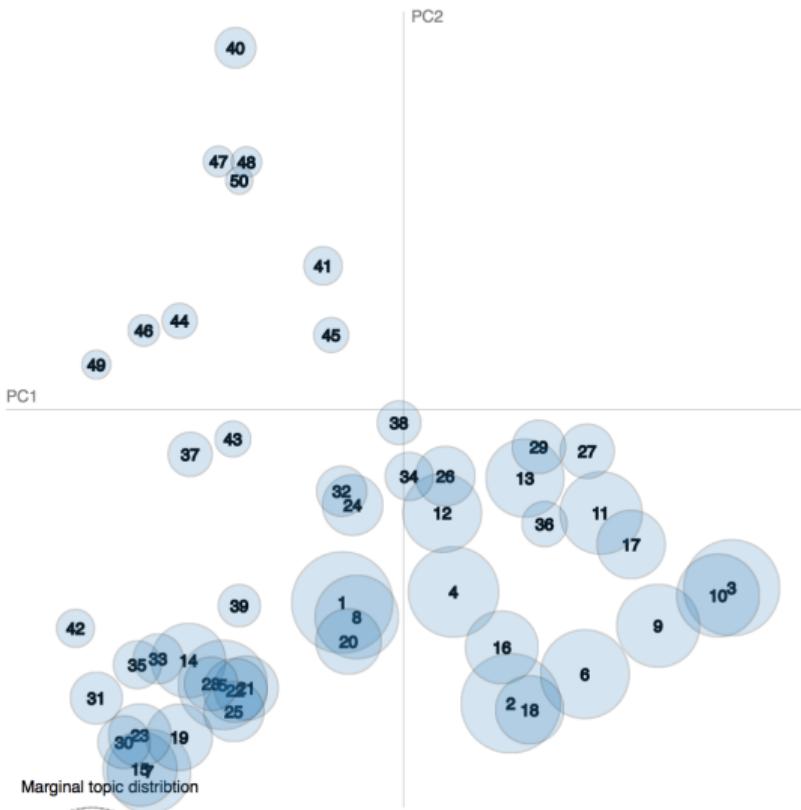
Selected Topic: 0    Previous Topic    Next Topic    Clear Topic

Slide to adjust relevance metric:(<sup>2</sup>)

$\lambda = 1$

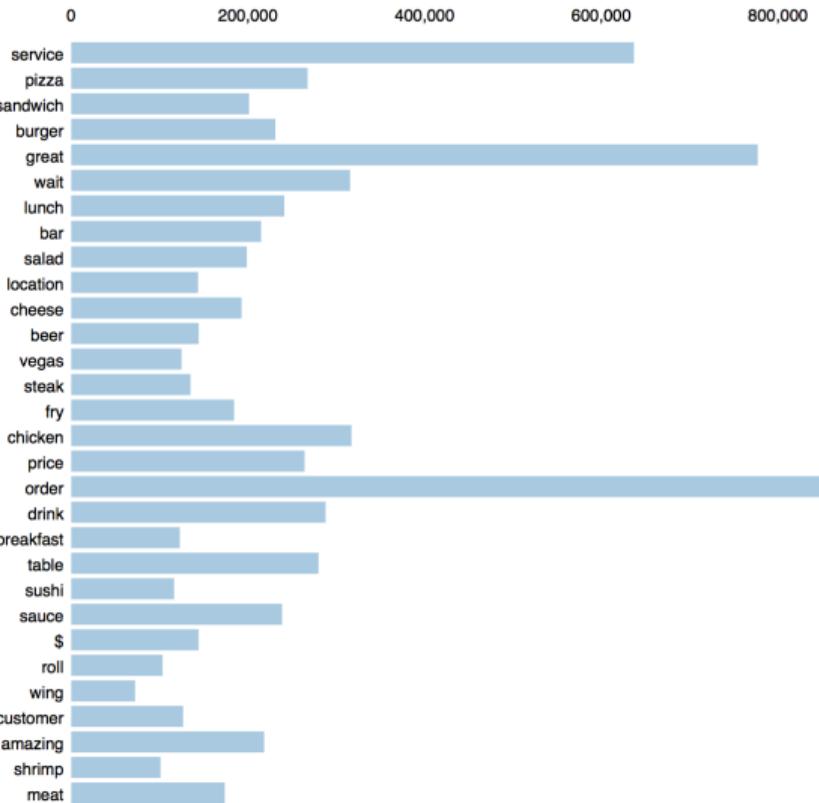
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

Top-30 Most Salient Terms <sup>(1)</sup>



Overall term frequency

# A Significantly Darker Box

- Selection criteria for language models
  - lack of information about initial training sets
  - **hard to evaluate except by considering output quality**
  - fine-tuning and retraining?

Token Classification

Transformers

PyTorch

TensorFlow

JAX

Rust

Safetensors

bert

Model card

Files and versions

xet

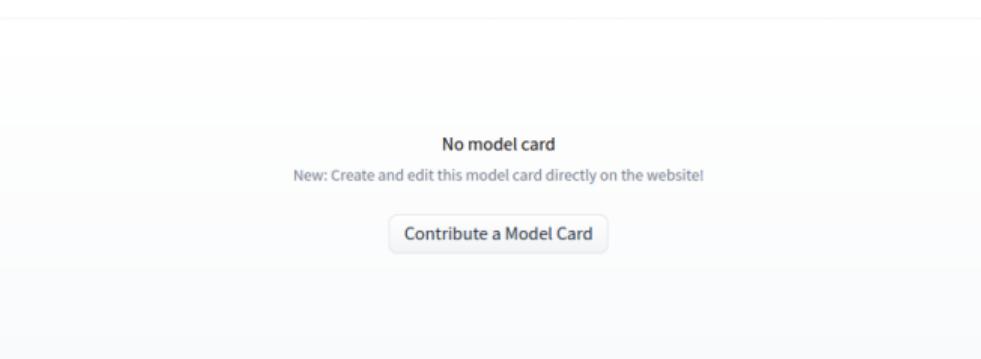
Community



Train

Deploy

Use this model



Downloads last month  
1,390,350



Safetensors

Model size 334M params

Tensor type F32

Files info

## Inference Providers

HF Inference API

Token Classification

Examples

Your sentence here...

Generate

View Code Snippets

Maximize

## Model tree for dbmdz/bert-large-cased-finetuned-conll03-english

Finetunes

1 model

Quantizations

1 model

## Spaces using dbmdz/bert-large-cased-finetuned-conll03-english 100

Zoe911/Transformers-NER

tchans123/resume\_enhancement\_tool

Guiliano/Reservoir-properties-calculator

sam6309/text\_to\_video

Gopi9177/NLP

zjunlp/KGEeditor

UjjwalVIT/Text\_analysis\_and\_metadata\_app

## Computer Science &gt; Computation and Language

[Submitted on 18 Jan 2018 ([v1](#)), last revised 23 May 2018 (this version, v5)]

# Universal Language Model Fine-tuning for Text Classification

Jeremy Howard, Sebastian Ruder

Inductive transfer learning has greatly impacted computer vision, but existing approaches in NLP still require task-specific modifications and training from scratch. We propose Universal Language Model Fine-tuning (ULMFiT), an effective transfer learning method that can be applied to any task in NLP, and introduce techniques that are key for fine-tuning a language model. Our method significantly outperforms the state-of-the-art on six text classification tasks, reducing the error by 18-24% on the majority of datasets. Furthermore, with only 100 labeled examples, it matches the performance of training from scratch on 100x more data. We open-source our pretrained models and code.

Comments: ACL 2018, fixed denominator in Equation 3, line 3

Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG); Machine Learning (stat.ML)Cite as: [arXiv:1801.06146 \[cs.CL\]](#)(or [arXiv:1801.06146v5 \[cs.CL\]](#) for this version)<https://doi.org/10.48550/arXiv.1801.06146> 

## Submission history

From: Sebastian Ruder [[view email](#)]

[v1] Thu, 18 Jan 2018 17:54:52 UTC (30 KB)

[v2] Mon, 14 May 2018 13:57:04 UTC (851 KB)

[v3] Tue, 15 May 2018 22:02:11 UTC (851 KB)

[v4] Thu, 17 May 2018 17:46:49 UTC (851 KB)

[v5] Wed, 23 May 2018 09:23:47 UTC (851 KB)

## Access Paper:

[View PDF](#)[TeX Source](#)[Other Formats](#)[view license](#)

Current browse context:

**cs.CL**[< prev](#) | [next >](#)[new](#) | [recent](#) | [2018-01](#)

Change to browse by:

[cs](#)[cs.LG](#)[stat](#)[stat.ML](#)

## References & Citations

[NASA ADS](#)[Google Scholar](#)[Semantic Scholar](#)

## 30 blog links

[\(what is this?\)](#)

## DBLP - CS Bibliography

[listing](#) | [bibtex](#)

Jeremy Howard

Sebastian Ruder

## Export BibTeX Citation

## Bookmark



# A Significantly Darker Box

- Selection criteria for language models
  - lack of information about initial training sets
  - **hard to evaluate except by considering output quality**
  - fine-tuning and retraining?
- Algorithmic details involve more free user choice
- Intermediates
  - context-dependent vectors for each token in the corpus (nope)
  - (sometimes) vectors for each document in the corpus (...maybe?)
  - final outputs like cluster assignments (sure, but...)

**A broader question: What exactly are we  
willing to outsource, and to whom?**

# Expertise

It is, of course, possible to build expertise in all of these questions ourselves.

But! Is it possible to do that and still be a subject-matter expert in our own research areas?

# Expertise

Have we reached the point at which we can no longer maintain the full collection of both **digital** and **humanities** skills? Is it time to consider that we might be forced to divide concerns?

# A Classic Problem

Collaborating with computer scientists tends to be difficult: the kinds of problems that we want to resolve *usually* aren't actually cutting-edge or innovative in CS terms.

We're boring!

So what we need to do, it seems, is hire research programmers. But precious few DH projects in philosophy are going to be big enough to justify an FTE of programming time.

**Taken together: a strong argument for  
constructing our own infrastructure in  
digital humanities.**

# **Building Infrastructure**

---

# CMU Library Labs (2020-2024)

Scott B. Weingart & Matthew Lincoln  
Fall 2019 (*edited January 2021*)

<http://dx.doi.org/10.1184/R1/13522718>

<b>Executive Summary</b>	<b>3</b>
<b>Personnel</b>	<b>5</b>
<b>Identity Statement</b>	<b>5</b>
Vision	5
Mission	5
Values	5
Goals	5
<b>Initiatives</b>	<b>6</b>
Incubator	6
Schedule & Load	6



# Project Incubator

1. Charter Phase (1 sem.): write a project charter and get everyone on board
2. Incubation Phase (1 sem.): execute on the project charter, build the project
3. Full Warranty Phase (1 sem.): one further semester of bug-fix work and production of a report detailing what was executed

# Project Incubator

4. Maintenance Phase (1 yr.): maintenance-only, ensuring that the project will continue to function in the same manner
5. Stewardship Phase (5 yr.+): rolling five-year maintenance contracts, only to ensure that the project is available (perhaps in a static or degraded form)

**Projects have to end. Plan now for the  
death of your creation.**

# Bring Money

Storage and basic computing expenses have always been significant, but still within the range of “normal” grant funding.

This gets worse with LLMs.

潮州焱



You cannot afford this on your grant.

**Major computing infrastructure has to be attached to something durable.**



NASA Discover HPC Cluster



Trinity College, Dublin

# All of a sudden...

...we're not just talking about getting up and running for a project. If you want to have DH make up a sustainable part of your career, you'll need to either be somewhere with this infrastructure, or build and maintain it yourself.

# Summing Up

---

## Two Main Worries

1. Can we keep up with the **pace of methodological change** without bringing in external collaborators?
2. Can we cultivate those collaborations (and pay for them) without building **durable infrastructure?**

(and what can you do if you don't have that infrastructure to begin with?)

# Questions?

charles@charlespence.net • <https://pencelab.be> •  [@pence@scholar.social](https://scholar.social)

