Episode Book

EPISODE #6032

Blending Datasets vs Merging Weights for Fine-Tuning

PUBLISHED	16 October 2026 at 17:02 UTC
DURATION	20:56
LISTEN	myweirdprompts.com/episode/dataset-blending-fine-tuning-guide-12789/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.5
SOURCES	18 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

If your use case is narrow enough to find training data for, blending multiple datasets into one fine-tune run can beat merging finished checkpoints. This episode walks through the full pipeline: why uniform or size-proportional sampling is usually suboptimal, the formatting tax that scales with every source you add, and why deduplication isn't optional — memorization scales superlinearly with repetition, and one dedup study saw up to ~20% perplexity improvement while cutting runtime nearly 28%. Then the tools: Hugging Face's concatenate and interleave, datamix, Oumi, LLMTune, and NeMo's code-switched data scripts — and why there's no single product that owns all four stages.

TRANSCRIPT



Corn

Daniel's been going back over the weight-averaging ground again. That was parameter space. He's now asking about the other direction entirely. If your use case is narrow enough that you can actually find datasets to train on, or you've already built several yourself, you blend the data and train once on the mixture. And the thing he's really circling is the wrangling. Getting inconsistent sources into one format, deduplicating them to some degree. Does anything exist that's built for exactly that, and has anybody actually shipped an ultra-specific fine-tune that pulled from several datasets instead of one.



Herman

The distinction he's drawing is the one worth keeping. Merging weights happens in parameter space. You've got two checkpoints that started from the same base, and you combine numbers without touching a single training example. Blending datasets is data space. You concatenate or interleave the corpora into one training set and you run a fine-tune on the mix.



Corn

Which means you need the data on hand, rather than just the finished checkpoints.



Herman

You need the data and you need a training run. But you get something merging can't give you. You decide the proportions before a single gradient step happens, rather than discovering afterward what the merge did.



Corn

And Daniel's premise is that if you can find the data, that's the more direct route.



Herman

It's more direct and it's messier, and the mess is the actual subject here. There's a survey on data mixture for large models that frames the whole thing cleanly. You've got k domains, and you're picking a weight vector w over them. It sits on a probability simplex. Non-negative, sums to one. It's just what fraction of the training mix each source gets.



Corn

Simple to state.



Herman

The finding is that those sampling proportions significantly move model performance. And that the obvious choices, uniform sampling or sampling proportional to how much data each source has, are both likely to be suboptimal. That's the part people don't expect. Proportional feels fair, and fair is not the same as good.



Corn

So the first real work is deciding the shape of the mix, and the formatting is just the pile of chores underneath it.



Herman

The chores are real, though. Start with the boring one, because it's the one that stops you in the first afternoon. Hugging Face's concatenate function on datasets only works if the datasets share the same column types. Same names, same types. If one source is called transcription and another is called text, it doesn't merge. It just fails.



Corn

That's a rename.



Herman

It's a rename until it isn't. One dataset has audio as a path, another has it as decoded arrays at a fixed sample rate, a third has it as bytes. Whisper-MCE is the honest illustration. That's the Mixed Cantonese-English work out of HKUST. The dataset was collected by a lot of students, and every student had their own naming rule for files, and their audio came in different formats. So before anything trains, there's a cleaning pass. That's not an edge case. That's what a dataset assembled by humans looks like.



Corn

Twenty people with twenty conventions. I've lived in one flat with one donkey for years and we still can't agree on where the bowls go.



Herman

Your bowls are in the wrong cupboard and you know it. But yes. The formatting problem isn't abstract. It's the tax you pay for every additional source you bring in, and it scales with the number of people who touched the data.



Corn

Is there a point where the tax is high enough that you should just walk away from a source?



Herman

There is, and the test is whether the source carries something the others don't. If you've got four hundred hours of read English and you're adding a fifth pile of read English, the marginal value is near zero and the formatting cost is real. But if that fifth pile is the only accented speech you have, you pay the tax. The tax buys coverage, not volume.



Corn

Now dedup. Daniel put it in as a parenthetical, "potentially some degree of deduplication." That phrasing makes it sound optional.



Herman

It's the opposite of optional, and there are two separate reasons, which is the part most people miss. The first is quality. Duplicated examples get over-weighted by the loss, the model spends capacity on things it already knows. The second reason is privacy, and that one is sharper. There's work from Kandpal and others on memorization scaling superlinearly with a sequence's count in the training data. A sequence that appears ten times isn't regenerated ten times more often than one appearing once. It's on the order of a thousand times more often.



Corn

A thousand. For ten.



Herman

Superlinear. Memorization doesn't scale with repetition, it accelerates with it. Which means if you blend three public corpora that overlap, and each one has the same transcript in it, you've built a corpus that will emit that transcript verbatim, and you didn't decide to do that. It happened in the concatenation.



Corn

And people blend their own private data with public data all the time.



Herman

Which is exactly the case where it bites. There's a dedup paper out of the federated learning side that puts numbers on the other end of it. At duplication levels between ten and thirty percent, deduplication bought them up to about twenty percent improvement in perplexity and cut runtime by nearly twenty-eight percent. You're deleting data and both the quality and the clock improve.



Corn

Deleting a quarter of the corpus and getting a better model out is a strange thing to hear.



Herman

It's the strongest argument in the whole pipeline. And notice what it costs you to do it. Nothing but a pass over the data. No extra training, no extra GPU hours.



Corn

Except the pass over the data is where it gets interesting, because the duplicate isn't the same string twice. It's the same audio under two filenames.



Herman

That's a real problem and the tools mostly don't see it. Exact string matching catches the transcript. It doesn't catch the same recording normalized to a different loudness, resampled, renamed, sitting in a different corpus with a slightly different orthography in the label. That's why dedup for audio is a separate discipline from dedup for text.



Corn

So text dedup is basically solved and audio dedup is the frontier.



Herman

Text dedup is solved in the sense that the tools exist and people use them. Audio dedup is where you're writing your own fingerprinting pass, and we'll get to what that actually looks like in a minute.



Corn

And this is where the weighting question sits, on top of the cleaned pile.



Herman

On top of it, and this is where the survey taxonomy gets useful. It splits the methods into offline and online. Offline means you pick the weights before training and hold them. Within that you've got heuristics, algorithm-based methods that estimate per-domain influence and reweight, and function-fitting methods that fit a curve to small runs and predict the mix that should work at scale. Online means the weights move during training, either by a min-max game where the model and the mixture alternate, or by fitting a mixing law and updating as you go.



Corn

And there's a newer one on the supervised fine-tuning side specifically.



Herman

It reframes mixing as an optimization problem. Minimize validation loss directly over the weights. The reported result is that the optimized weights land on par with a full grid search, with per-domain loss only about two-thirds of a percent higher on average.



Corn

Which sounds like a rounding error until you remember what the grid search costs.



Herman

A grid search over four sources at ten percent granularity is thousands of training runs. The optimization gets you there in a fraction of the compute. That's the trade. Same quality, no grid.



Corn

And the online methods, the ones that move the weights mid-run. What do they cost you?



Herman

Complexity, mostly. You're now maintaining two moving parts instead of one. The model is learning and the mixture is learning, and if the mixture's update rule is badly tuned you can chase your own tail. But the payoff is that you never have to commit to a weight vector up front, which matters when you don't know the right answer and can't afford to search for it.

L

Larry

Are you tired of explaining to friends and family that your fine-tune still cannot handle a bilingual phone call? You are not alone. Introducing Blend-O-Matic Pro, the world's first fully automatic dataset blender. Just pour in all your datasets, any format, any language, any decade, and press the one button. Blend-O-Matic Pro silently harmonizes every column name, removes every duplicate it can find, and promises a model that speaks every language perfectly, including three that do not exist. But wait, there's more. Order in the next eight minutes and we will include the Curriculum Scheduler attachment, which reorders your data by difficulty while you sleep. Side effects may include unexpected fluency, spontaneous code-switching, mild confusion about which of your own datasets were private, and a model that answers questions you never asked. Blend-O-Matic Pro is not responsible for any perplexity that results. Results not typical. Results not verified.

L

Larry (continued)

Results, frankly, not expected.



Herman

So with the pile cleaned, you still have to decide the weights, and that's where the tooling question lands. Daniel asked whether dedicated tools exist, and the honest answer is that there isn't one end-to-end product. There's a pipeline, and you assemble it.



Corn

Walk me through the pieces.



Herman

Bottom layer is the datasets library itself. concatenate stacks datasets with matching schemas. interleave alternates between sources and takes a probabilities argument, so you can weight each one, and a stopping strategy that decides what happens when one source runs dry. First exhausted stops the lot the moment any source is spent. All exhausted keeps going and reshuffles the finished ones.



Corn

That choice matters a lot more than it looks like it does.



Herman

It decides what your last five percent of training looks like. If your smallest source exhausts early and you stop, you never see the big one again. There's a worked notebook in Hugging Face's community events repo that builds a Spanish blend out of Common Voice, VoxPopuli, Multilingual LibriSpeech and FLEURS. The stated motivation is exactly the low-resource case. Any one of those alone might not have enough data, so you interleave all four.



Corn

So the baseline works and it's four lines of code.



Herman

And then you're doing the weighting by hand. Which is why people reach for the layer above. There's datamix, which is a zero-dependency tool that profiles your datasets, generates mix recipes, schedules a curriculum, allocates token budgets and cleans the data. There's Oumi, which gives you a dataset framework pulling from local files, the Hub, or cloud storage, with quality scoring and PII detection built in. LLMTune is similar in shape. And on the synthetic side, NeMo ships scripts that build code-switched data by concatenating monolingual audio.



Corn

That last one is quietly the interesting one, because it isn't blending datasets you found. It's manufacturing a source.



Herman

It's manufacturing the exact source you were missing. And there's a vendor feature that points the same direction. Amazon Nova's supervised fine-tuning lets you mix your own samples with their proprietary training data as part of the job.



Corn

So the pipeline is datasets for assembly, something for dedup, a weighting method on top, and a framework to run it.



Herman

And no single product that owns all four stages. The pattern is a stack, and you own the seams between the layers.



Corn

Does Daniel still have that Parakeet setup running, the two separate tunes?



Herman

Hebrew-English code-switching and the technical vocabulary one, yes. Which maps onto this almost exactly, and there's a small wrinkle. This two-way ASR split he described, one tune for code-switching and one for main vocabulary, that maps onto the weight-merging literature. Task Arithmetic, TIES, DARE. Nobody's published that exact two-way split for ASR that I can find.



Corn

The closest analogue?



Herman

GLoRA, out of the ICASSP workshop. Korean-English code-switching, and it combines fine-tuned monolingual and multilingual ASR models using gated low-rank adaptation. So it's merging checkpoints, but it's the code-switching version of the idea.



Corn

Which is the parameter-space route again, wearing a different hat.



Herman

It is. Now let's go to what people have actually built on the data side, because there are real results and one of them is surprising.



Corn

Start with the Cantonese-English one, since we've already been in its dataset.



Herman

Whisper-MCE. They fine-tuned Whisper on a Mixed Cantonese-English dataset they generated themselves, using a multi-agent framework, and the media and many datasets came in from the HKUST collection we talked about. The headline is fourteen point two eight percent mix error rate. That's thirty-five percent lower than the original Whisper on the same task, and twelve point six one percent character error rate on Common Voice Cantonese.



Corn

So a model that was never trained for the mix, plus a purpose-built blend, plus one training run.



Herman

And the wrangling was the work. The model architecture is the boring part of that project. The interesting part is the cleaning pass on student-collected audio.



Corn

Now the surprising one.



Herman

Catalan-Spanish code-switching, out of Interspeech. They fine-tuned Whisper Large version three on a blend of three things. Four hours of real code-switched speech from a Catalan TV station. A hundred and five hours of Common Voice. And seventeen hours of synthetic code-switched data they generated by stitching monolingual clips together.



Corn

Three sources, wildly different sizes.



Herman

And the winner was the synthetic blend. Fourteen point four eight percent word error rate on the parliamentary test set, which beat an approach that used a hundred hours of concatenated read speech.



Corn

Seventeen hours beat a hundred.



Herman

Seventeen synthetic beat a hundred real. The paper calls it a remarkable finding in its own words, and it's right to. It inverts the intuition that more real data always wins. What won was the right kind of data, generated to match the exact distribution they cared about, rather than a large pile of read speech that sounds nothing like a bilingual conversation.



Corn

That's the part I'd sit with. Daniel's whole framing is "if you can find datasets to train on." And the finding is that sometimes finding isn't the move.



Herman

Generating is the move. If you know your target distribution exactly, you can synthesize to it. And the Nemo scripts we mentioned exist because that's now a normal step, not a research stunt.



Corn

What else is out there on the multi-dataset ASR side?



Herman

There's a Whisper Small fine-tune for Egyptian Arabic code-switching that pulls from a diverse set of sources specifically to capture the switch points. And then the structural result, which is the one I keep coming back to. The survey's Mixture of Data Experts



Herman (continued)

method trains one expert per domain and then blends them.



Corn

So it does both. It blends the data and then merges the models.



Herman

One expert per domain, then a merge across domains. The two routes aren't competing. They're two ends of the same pipe. You can blend datasets to train better experts, and then merge those experts, and you've used both layers.



Corn

Which is the answer to Daniel's original question and the answer to this one, and they turn out to be the same shape.



Herman

With one trap worth naming. The survey flags that optimal mix weights may not transfer. Not across model architectures, not across tokenizers, and not from a small proxy model to a larger target model.



Corn

So the standard move, tune the blend on something small and cheap, then scale up...



Herman

Is flying blind. You tuned on a proxy and the proxy's optimal mix is not the target's. Which is a brutal practical problem, because the whole appeal of tuning on a small model is that the big one costs too much to sweep.



Corn

You either pay for the sweep at scale, or you accept that your weights are a guess.



Herman

Or you use one of the online methods that adapts weights during the run, which is effectively paying for the tuning inside the training job instead of beside it.



Corn

Hm. That's a real fork in the road for anyone doing this at home.



Hilbert

The thing nobody ever says about deduplication is that the duplicates that hurt are never in the same file.



Hilbert

Everyone runs a dedup pass and feels finished. You hash the transcripts, you drop the repeats, you move on. That catches the lazy case. The case that gets you is one recording that lives in four places under four different names.



Corn

Same audio, four filenames.



Hilbert

Same audio. I found it because of a folder. Twenty-six thousand files, all of them wav, all of them with names that were just numbers. I'd been told it was three hundred hours. It was not three hundred hours.



Herman

How much was it?



Hilbert

About ninety. The rest was the same recordings, resampled, renamed, some of them with a different room tone because somebody had run them through a noise gate in between. Eight thousand six hundred files were one recording of a man reading a phone book. Different names every time.



Corn

One man, reading a phone book, eight thousand times.



Hilbert

He was a volunteer. They kept re-uploading his session because it was the cleanest audio they had. Somebody would clip a segment, rename it, put it back in the pot, and the next person would do the same. Nine rounds of that and the algorithm can't see it. The names don't match. The durations don't match, because of the clipping. Only the audio matches, and you only catch that if you fingerprint the waveform itself.



Herman

Perceptual hashing rather than string hashing.



Hilbert

You run the fingerprint, you cluster, and you find out your dataset is one guy and eighty-nine other people. So I deduplicated it down to the ninety hours and gave it back, and they asked me why I'd deleted two-thirds of their data.



Corn

What did you tell them?



Hilbert

That it had already been in there twice. They didn't like it. It went in anyway. It's in three models now, and it's in a Japanese one, which I did not expect, because nobody asked me about the Japanese one.



Herman

So the practical version of the memorization result is that you caught it by hand, with a fingerprint.



Hilbert

You catch it with a fingerprint or you don't catch it. Twenty-six thousand files, ninety hours. The tokenizer on the Japanese one still splits the man's name in three.



Corn

Thank you, Hilbert.



Herman

The closing thought I keep landing on is the weight-transfer problem. You can tune a mix on a small proxy, and the survey says plainly that the optimum won't carry to the target. So the honest position for anyone doing this seriously is that the sweep at scale isn't optional, or you go online and let the weights move mid-run.



Corn

The other shift, which that Catalan result points at. If seventeen hours of synthetic data beat a hundred hours of read speech, then the problem stops being "find more data." It becomes "generate the right data," and the wrangling moves from formatting and dedup to controlling the quality of what you manufactured.



Herman

The two layers are complementary, in the end. Blend the data to build better experts, merge the experts, and the interesting work is at the seam between them. That Mixture of Data Experts result does both in one pipeline, and I suspect that's where this lands eventually.



Corn

The cutting-room floor item, since it fits: mergekit's out-of-core merging runs on a CPU or as little as eight gigabytes of VRAM. The merging side of this is cheap hardware. The expensive half is the data.



Herman

Which is backwards from how people talk about it.



Corn

Thanks to Hilbert Flumingtop, our producer, who is still at the desk.



Herman

If you want more of this, try episode six, How To Fine Tune Whisper. This has been My Weird Prompts.



Corn

If you want to send us a prompt, there's a Telegram bot. Send us your own prompt on Telegram at t dot me slash MWP listener bot. We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **mergekit repository (Arcee AI) [primary]**
README, merge_methods.md, commit 9eeb539
https://github.com/arcee-ai/mergekit

2. **Hugging Face `datasets` [primary]**
Process docs (concatenate/interleave)
https://huggingface.co/docs/datasets/process

3. **Hugging Face `datasets` combine.py source [primary]**
https://github.com/huggingface/datasets/blob/main/src/datasets/combine.py

4. **HF community-events [primary]**
interleave_streaming_datasets.ipynb (Whisper fine-tuning event)
https://github.com/huggingface/community-events/blob/main/whisper-fine-tuning-event/interleave_streaming_datasets.ipynb

5. **Whisper-MCE repository (ICASSP 2025) [primary]**
https://github.com/shelton1013/whisper_mce

6. **datamix [primary]**
dataset mixing & curriculum optimizer
https://github.com/stef41/datamix

7. **Oumi OSS datasets docs [primary]**
https://oumi.ai/docs/en/latest/resources/datasets/datasets.html

8. **Amazon Nova SFT data mixing docs [primary]**
https://docs.aws.amazon.com/nova/latest/userguide/nova-forge-sft.html

9. **Rethinking Data Mixture for Large Language Models: A Comprehensive Survey**
arXiv:2505.21598
https://arxiv.org/html/2505.21598

10. **Data Mixing Optimization for Supervised Fine-Tuning of LLMs**
arXiv:2508.11953 (2025-08-16)
https://arxiv.org/abs/2508.11953v1

11. **Model Merging in LLMs, MLLMs, and Beyond**
arXiv:2408.07666v4
https://arxiv.org/html/2408.07666v4

12. **NVIDIA Technical Blog**
An Introduction to Model Merging for LLMs
https://developer.nvidia.com/blog/an-introduction-to-model-merging-for-llms/

13. **Merging in a Bottle: Differentiable Adaptive Merging (DAM)**
arXiv:2410.08371
https://arxiv.org/html/2410.08371

14. **Deduplicating Training Data Mitigates Privacy Risks in Language Models**
arXiv:2202.06539v3
https://arxiv.org/abs/2202.06539v3

15. **Privacy-Preserving Data Deduplication for Federated Learning of LMs**
arXiv:2407.08152v2
https://arxiv.org/abs/2407.08152v2

16. **Optimizing ASR for Catalan-Spanish Code-Switching**
arXiv:2507.13875
https://arxiv.org/html/2507.13875

17. **Gated Low-rank Adaptation for personalized Code-Switching ASR**
arXiv:2406.02562v1
https://arxiv.org/abs/2406.02562v1

18. **IbrahimAmin/code-switched-egyptian-arabic-whisper-small**
https://huggingface.co/IbrahimAmin/code-switched-egyptian-arabic-whisper-small