WHITE PAPER

Hebrish model release 1: Jachnun v1.0 and Sabich v1.0

The first stable Hebrish models: two fine-tunes of NVIDIA Parakeet for English-Hebrew code-switched dictation. How they were trained, exported and benchmarked, what they get right, and what the numbers do not show. With a note on Shakshuka, the streaming line, still in progress.

AUTHOR
Daniel Rosehill

DATE
21/10/2026

Models: `Hebrish/Jachnun` (run `r6-en30`) and `Hebrish/Sabich` (run `r7-all`). Replaces the “Hebrish v1.0” white paper of 2026-10-19 and its 2026-10-20 addendum.

Abstract
English speakers in Israel drop Hebrew words into English. English speech recognition has rarely heard those words and writes the nearest English-sounding thing. Hebrish model release 1 is the project’s first stable set of models: two full fine-tunes of NVIDIA Parakeet TDT 0.6B v2, each taught to write every Hebrew word in one canonical romanized spelling (*makolet*, *Bituach Leumi*). **Jachnun v1.0** learned from 339.2 minutes of code-switched read speech plus about 30% English. **Sabich v1.0** learned from the same kind of recordings plus 187.4 minutes of tech and task recordings, and is the model the Hebrish apps recommend. On the public English-Hebrew code-switching benchmark (test set v1, clean subset, 84 clips) Jachnun scores 73.3% Hebrew term recall, 70.1% on terms absent from training and 3.3% WER, and Sabich 73.8%, 68.4% and 3.3%, against 32.6%, 32.5% and 9.5% for stock Parakeet v2. On 422 held-out clips they are level on code-switching (5.42% and 5.30% WER, stock 11.82%), Sabich is better on tech dictation, and plain English did not get worse. A third line, **Shakshuka**, the same recipe on NVIDIA’s Nemotron streaming model, is a proof of concept and not part of this release. All audio, training and test, is one speaker’s voice, so these numbers rank models on that speaker only.

73.8%
Sabich term recall, public benchmark (stock v2: 32.6%)

73.3%
Jachnun term recall; 70.1% on terms never in training

3.3%
WER, public benchmark, both models (stock v2: 9.5%)

1.54%
Sabich tech-dictation WER (Jachnun 1.74%, stock 2.51%)

1. Release 1 at a glance

Hebrish is the project, not a model: English dictation that writes the Hebrew words correctly, free apps for Android, Ubuntu and Windows, and the open models and data behind them. Its model lines

are named after Israeli foods. **Model release 1** is the first stable release of the two lines built on Parakeet; each is at version **v1.0**. Before it came five proof-of-concept checkpoints, PoC-1 to PoC-5 (published on Hugging Face as v1 to v5 of `danielrosehill/Parakeet-TDT-0.6B-v2-English-Hebrew-Codeswitching`).

Model	Base	Trained on	Run · date	In the apps
Sabich v1.0	**Parakeet TDT 0.6B v2**	**Code-switching plus tech and task recordings (623 min)**	`r7-all` · **2026-10-20**	**Recommended (since 2026-10-21)**
Jachnun v1.0	**Parakeet TDT 0.6B v2**	**Code-switching plus English (483 min)**	`r6-en30` · **2026-10-19**	**Second choice**
Shakshuka PoC-2	Nemotron Speech Streaming EN 0.6B	Jachnun's recordings and recipe	`r6-nemotron-en30` · 2026-10-19	Not yet (Section 9)

Retrains on more recordings will be Jachnun v1.1, v1.2 and so on (likewise Sabich). Shakshuka's first retrain will be Shakshuka v1.0.

2. The problem

Many English speakers in Israel speak English with Hebrew words in it: *"grab some bamba from the makolet"*, *"I have to go to Bituach Leumi"*. This is code-switching. English dictation models were trained on English and have rarely or never heard these words, so they substitute the nearest English-sounding word: *Macaulay* for *makolet*. Often the one word that carries the meaning is the one that comes out wrong.

Hebrish keeps English dictation as it was and writes the Hebrew words correctly. "Correctly" needs a definition, because Hebrew words in English text have no single agreed spelling. Hebrish uses one canonical romanized spelling per word, from a public spelling lexicon (`terms.csv`), and writes Hebrew in Latin letters only. Parakeet's tokenizer is a 1,024-token SentencePiece BPE with cased tokens and punctuation but no Hebrew characters, so Hebrew script was never an option; romanization also suits the use case, which is English text.

The base model is NVIDIA Parakeet TDT 0.6B v2 (CC-BY-4.0), a 0.6B-parameter FastConformer TDT model. It is small enough to run on a laptop CPU and a phone, which is where dictation happens.

3. Dataset

Training data is Read-Aloud-Voice-Sentences (CC-BY-4.0; access on request since 2026-10-20, see hebrish.app/datasets). Every clip is **one speaker** (Daniel Rosehill, Irish-born, English-speaking, living in Jerusalem) reading a prompted sentence aloud, mostly on one phone, as 16 kHz mono audio with a cased, punctuated transcript and a list of its target terms. Code-switched sentences carry the Hebrew terms; plain-English and task sentences are recorded in the same voice. The table is the split at revision `550bfcaa0f84`, the one Jachnun was trained and first evaluated on.

Split at `550bfcaa`	Clips	Minutes	Use
Code-switching train (`cs_train`)	3,224	339.2	Training
Code-switching validation (`cs_val`)	218	22.6	Best-epoch selection
Own-English pool	751	95.3	Training (all of it)
LibriSpeech-PC replay pool	900	190.7	Training (48.5 min drawn)
Code-switching test (`test_cs`)	387	43.9	Evaluation
Own-English test	73	9.2	Forgetting check
LibriSpeech test-clean	200	24.6	Forgetting check (other speakers)
Long-form test	15 files	44.6	Long recordings

Held-out terms

The split is **term-disjoint for a held-out list**, not for every term. 362 terms are on the held-out list at this revision; a sentence containing any of them goes to test, so those terms never appear in training. A test sentence can also contain other terms that are not on the list, and those can appear in training. The evaluation therefore reports two numbers: recall on held-out terms (293 occurrences of 118 terms in `test_cs`), which measures generalisation, and recall on test terms also seen in training (278 occurrences), which measures learning. Only the first is evidence that the model writes a word it was never taught.

4. Two recipes from one base

Both models are trained from stock `nvidia/parakeet-tdt-0.6b-v2`; neither is trained on top of the other. Only the blend differs. Jachnun learns from code-switched speech and the speaker's plain English. Sabich, named after the pita with a bit of everything, learns from every task the speaker recorded.

Figure 1. One base, two recipes. Minutes are audio in each training blend.

Training blend (minutes)	Jachnun v1.0 (`r6-en30`)	Sabich v1.0 (`r7-all`)
Code-switching	339.2	373.1
Speaker's other recordings	95.3 (own English)	187.4 (tech and task)
LibriSpeech-PC replay	48.5	62.5 (10%)
Total	483.0 (4,202 rows)	623.0

Sabich's 187.4 minutes of task recordings cover tech vocabulary, brand names, acronyms, numbers and currency, Irish-English, disfluencies, self-corrections and other tasks; with the code-switching that is 560.5 minutes of the speaker's own recordings.

Why 30% English

A dictation model that learns Hebrew words but loses English is no use. Jachnun's English share was targeted at 30% and came out at 29.8%: the speaker's own English teaches that plain English in this voice should stay plain English, and LibriSpeech replay keeps other voices. The share follows the code-switching and continual-learning literature: fine-tuning only on new-domain data is what causes catastrophic forgetting, and mixing in the original domain plus some replay of general data is the usual remedy. Sabich keeps the replay at 10%; its task recordings are mostly English in the same voice. Section 8 checks whether it worked.

Why a cumulative full fine-tune, not continued training

Each release starts from stock Parakeet and trains on every take recorded so far. Continuing the previous fine-tune on only new data is what causes catastrophic forgetting, and continuing on all data gives old rows another round of epochs each release, so drift compounds. A run takes under an hour on one GPU, so nothing is saved by continuing. Training and benchmarking are separate steps: the best epoch is chosen on validation data (part of train, never test), and the benchmark runs afterwards on test.

5. Training

Setting	Jachnun v1.0	Sabich v1.0
Framework	NeMo 2.7.3 (`nemo_toolkit[asr,cu12]`), torch 2.11.0, pinned in a Modal image	
Start point	`nvidia/parakeet-tdt-0.6b-v2`, tokenizer kept	
Optimiser	AdamW, lr 3e-5, betas 0.9 / 0.98, weight decay 1e-3	
Schedule	Cosine annealing, 10% warmup, min lr = lr / 20	
Epochs / batch	20 epochs, batch size 16, bf16 mixed precision, seed 7	
Selection	Epoch 18 of 20, validation WER 5.0% (raw, 0.0497)	Epoch 19 of 20, validation WER 4.63%
Hardware / time	One NVIDIA L40S (48 GB) on Modal, 41.0 minutes	One NVIDIA L40S on Modal, 52.1 minutes

From proof of concept to release 1

Checkpoint	Run	CS min	Released	Benchmarked
PoC-1	`pilot-en30`	77.4	2026-10-15	Yes
PoC-2	`r2-en30`	120.1	2026-10-15	Yes
PoC-3	`r3-en30`	196.1	2026-10-18	Yes
PoC-4	`r4-en30`	269.3	2026-10-18	No
PoC-5	`r5-en30`	294.2	2026-10-18	No
Jachnun v1.0	`r6-en30`	**339.2**	**2026-10-19**	**Yes, 2026-10-19**
Sabich v1.0	`r7-all`	**373.1**	**2026-10-20**	**Yes, 2026-10-20**

PoC-1 to PoC-5 used Jachnun's recipe on less data. Jachnun v1.0 is the sixth checkpoint of that line; it is called v1.0 because it is the first release the project stands by. Sabich v1.0 is not a later checkpoint but a second recipe from stock.

6. Exports

Both models are published in the formats on-device runtimes use (Hugging Face, access on request). Jachnun's exports were checked on five clips against fp32 ONNX before any benchmark.

Format	Repositories	Runs in	Check against full precision
`.nemo`	`Hebrish/Jachnun`, `Hebrish/Sabich`	NeMo (PyTorch)	Reference
sherpa-onnx int8	`Hebrish/Jachnun-sherpa-onnx-int8`, `Hebrish/Sabich-sherpa-onnx-int8`	sherpa-onnx, the Hebrish apps	Jachnun 4/5
ONNX fp32 and int8	`Hebrish/Jachnun-ONNX`, `Hebrish/Sabich-ONNX`	onnx-asr, Voxtype, the in-browser demo	Jachnun int8 4/5; Sabich fp32 5/5, int8 3/5
GGUF F16, Q8_0, Q4_K_M	`Hebrish/Jachnun-GGUF`, `Hebrish/Sabich-GGUF`	transcribe.cpp, Handy, Offline Voice Input	Jachnun F16 5/5, Q8_0 5/5, Q4_K_M 4/5
MLX	`Hebrish/Jachnun-MLX`, `Hebrish/Sabich-MLX`	parakeet-mlx (Apple Silicon), with `--fp32`	10 clips: Jachnun 8/10, Sabich 9/10

Jachnun's int8 misses are one word: *tzatzumim* where fp32 wrote *satzumim* the reference is *tza'atzuim*. Sabich's int8 misses were one capital letter and one comma; its MLX miss, one comma. In bf16, parakeet-mlx's default, Jachnun decoded one clip as unknown tokens, hence fp32. The benchmark numbers below are for the `.nemo` in PyTorch, not the quantized exports.

7. Evaluation

Two test sets answer different questions.

Public benchmark

The English-Hebrew code-switching ASR benchmark (dataset `Hebrish/English-Hebrew-Codeswitching-ASR-Benchmark`, test set v1, frozen) has 145 clips. The headline **clean subset** is 84 phone recordings (11.8 min) of sentences in no published training set, with 172 Hebrew term occurrences (139 distinct). A separate **English control** subset of 24 clips is the same speaker reading English only. Jachnun was scored in round 3 (2026-10-19), Sabich in round 4 (2026-10-20, pinned at Hugging Face revision `640c1483`).

- **Term recall**: share of term occurrences whose canonical spelling appears in the transcript as a whole word, case-insensitive, hyphen and space interchangeable. **Unseen-term recall** restricts it to terms absent from the public training split (117 of the 172 occurrences).

- **WER**: lowercased, hyphens to spaces, punctuation removed except in-word apostrophes (`jiwer` 3.1.0).
- **Contamination check**: none of the 145 benchmark clips, by id or by text, is in Jachnun's training manifest (`runs/r6-en30/train.jsonl`, 4,202 rows), Sabich's training or validation data, or the Shakshuka run's.

Held-out splits

The project's own test split is larger. Two revisions are used, and numbers compare only within one: `550bfcaa0f84` (387 code-switched clips, Section 3), on which stock v2, PoC-3, Jachnun and Shakshuka PoC-2 were scored on 2026-10-19; and the newer `f956dbb184c2` (round R7: 422 code-switched clips, 84 own-English clips of which 57 have no Hebrew, 20 tech-dictation clips, 200 LibriSpeech clips, 17 long-form files), on which stock v2, Jachnun and Sabich were scored on 2026-10-20. WER is normalised.

8. Results

Public benchmark

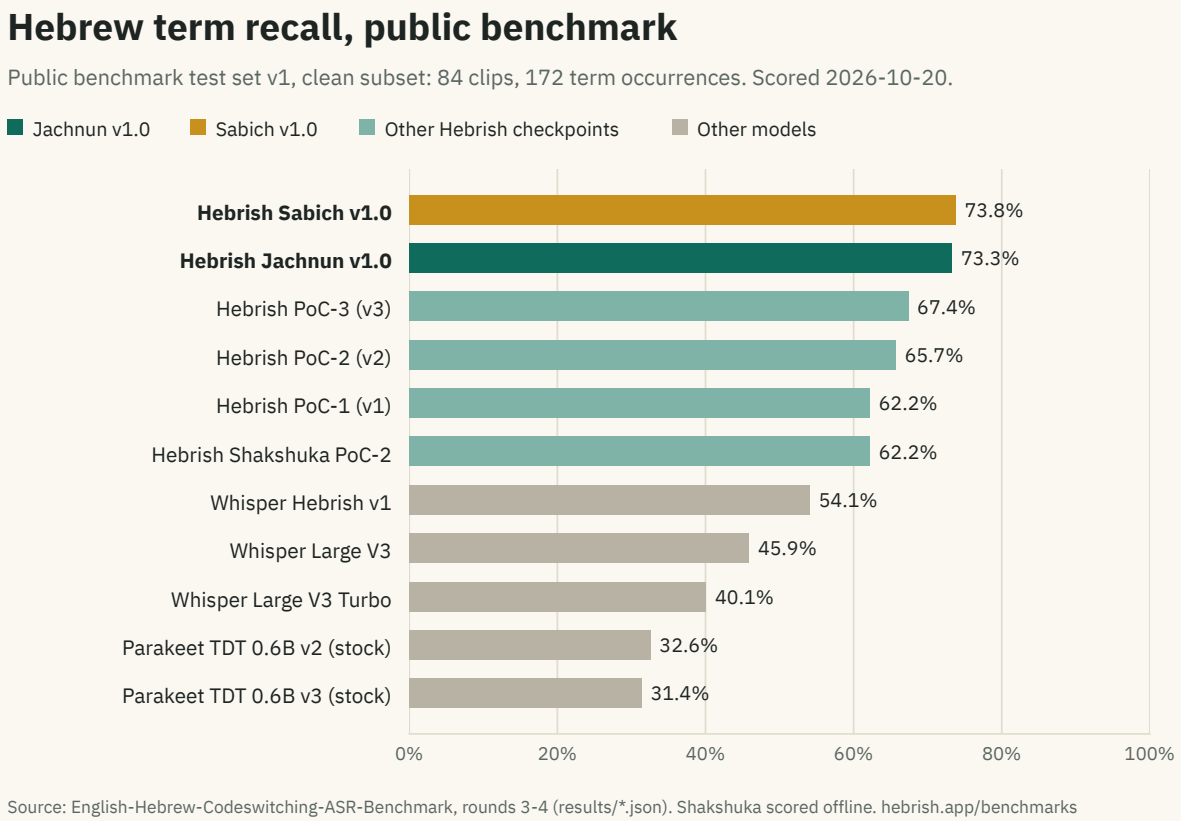

Figure 2. Hebrew term recall on the public benchmark, clean subset, all eleven scored models.

Model (public benchmark, clean subset)	Term recall	Unseen recall	WER	English control WER
Hebrish Sabich v1.0	**73.8%**	**68.4%**	**3.27%**	**1.99%**
Hebrish Jachnun v1.0	**73.3%**	**70.1%**	**3.32%**	**2.17%**
Hebrish PoC-3	67.4%	63.2%	4.0%	2.5%
Hebrish PoC-2	65.7%	60.7%	4.2%	2.4%
Hebrish PoC-1	62.2%	61.5%	4.6%	2.5%
Hebrish Shakshuka PoC-2 (offline)	62.2%	54.7%	4.7%	2.5%
Whisper Hebrish v1	54.1%	56.4%	6.5%	4.7%
Whisper Large V3	45.9%	46.2%	7.5%	2.4%
Whisper Large V3 Turbo	40.1%	39.3%	8.9%	2.4%
Parakeet TDT 0.6B v2 (stock)	32.6%	32.5%	9.5%	2.7%
Parakeet TDT 0.6B v3 (stock)	31.4%	31.6%	10.5%	2.7%

On 172 occurrences, a 2 to 3 point difference in recall is about 4 to 5 hits and within noise. Sabich against Jachnun is one hit of 172 on recall and two of 117 on unseen recall.

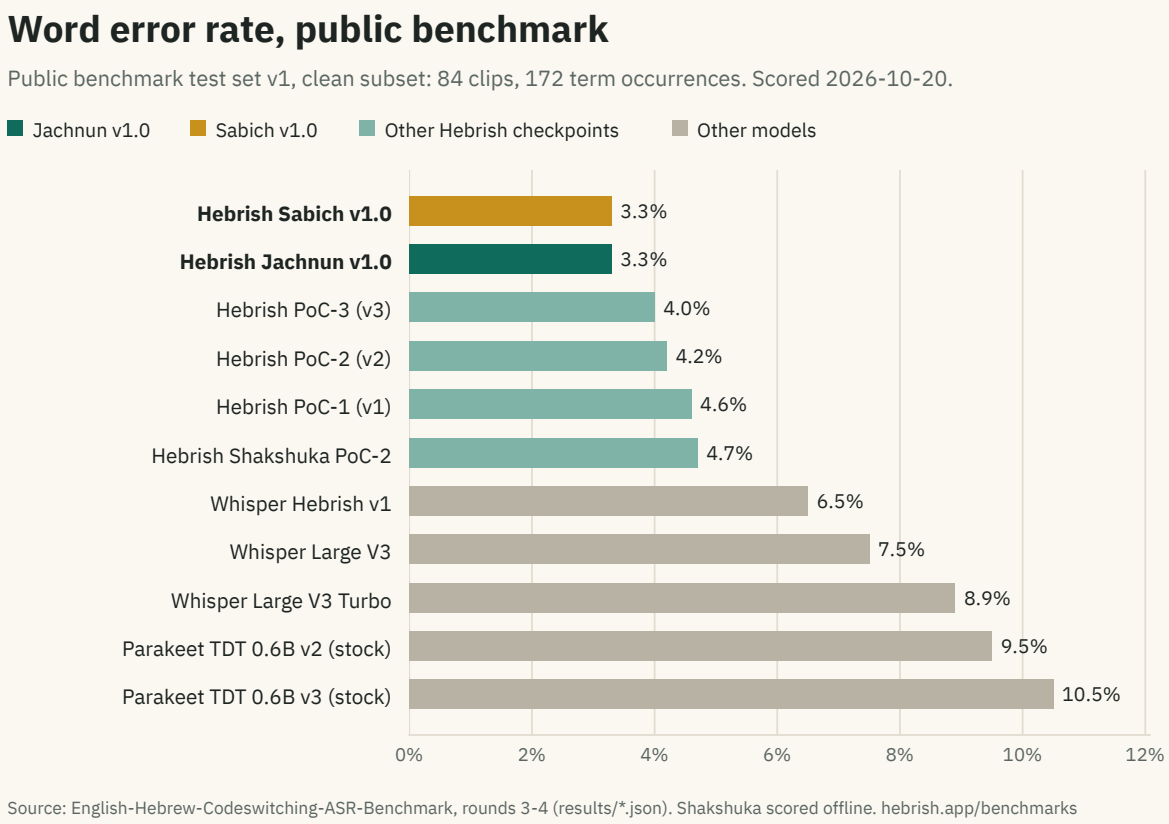

Figure 3. Word error rate on the public benchmark, clean subset, all eleven scored models.

Jachnun on the held-out split (`550bfcaa`)

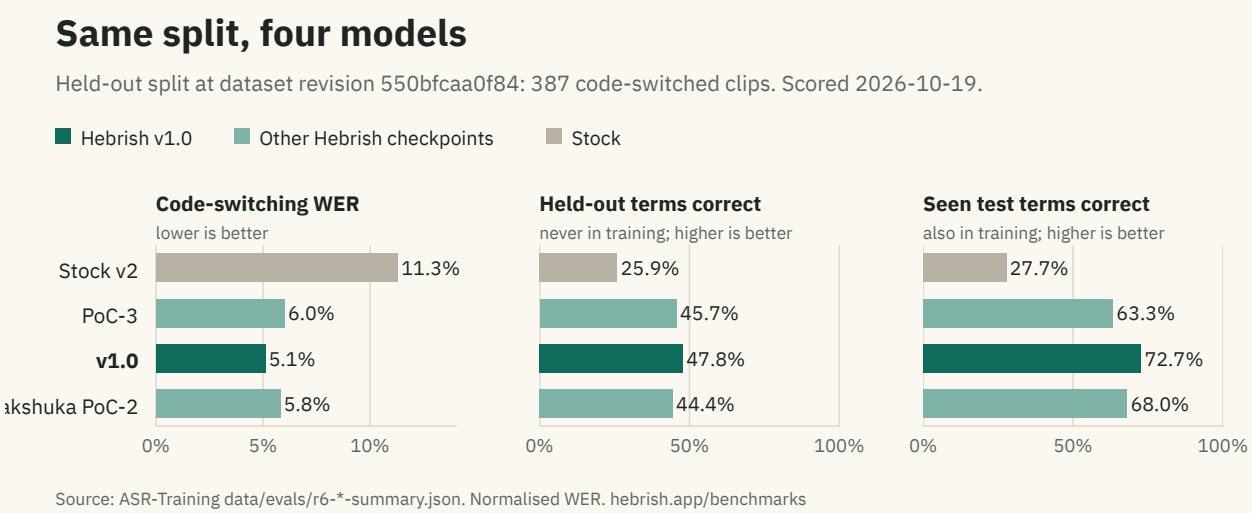

Figure 4. Stock v2, PoC-3, Jachnun v1.0 and Shakshuka PoC-2 on the same held-out split.

`550bfcaa`	CS WER	Held-out terms (293)	Seen terms (278)	Own EN	LibriSpeech	Long-form
Stock Parakeet v2	11.3%	25.9%	27.7%	3.1%	1.2%	9.6%
Hebrish PoC-3	6.0%	45.7%	63.3%	2.3%	1.0%	5.8%
Jachnun v1.0	**5.1%**	**47.8%**	**72.7%**	**1.7%**	**1.2%**	**5.0%**
Shakshuka PoC-2 (offline)	5.8%	44.4%	68.0%	3.3%	2.0%	5.8%

Jachnun and Sabich on the same clips (`f956dbb1`, round R7)

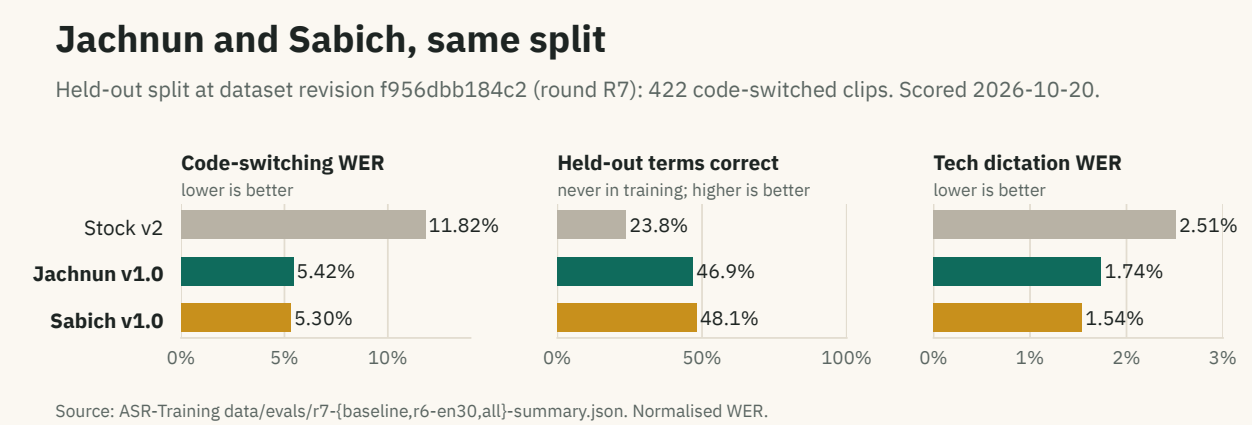

Figure 5. Stock v2, Jachnun v1.0 and Sabich v1.0 on the R7 held-out split. New in this edition.

`f956dbb1`	CS WER	Held-out terms	Own EN	Hebrew-free	Tech	LibriSpeech	Long-form
Stock Parakeet v2	11.82%	23.77%	2.81%	1.79%	2.51%	1.23%	10.34%
Jachnun v1.0	**5.42%**	**46.91%**	**1.56%**	**1.23%**	**1.74%**	**1.20%**	**5.84%**
Sabich v1.0	**5.30%**	**48.15%**	**1.62%**	**1.23%**	**1.54%**	**1.30%**	**5.77%**

Held-out terms: share of held-out-list term occurrences written correctly (higher is better). Every other column is WER (lower is better). Hebrew-free is the 57-clip subset of own English with no Hebrew words. This revision is newer and larger than `550bfcaa`, so Jachnun's numbers differ from the table above.

English did not get worse

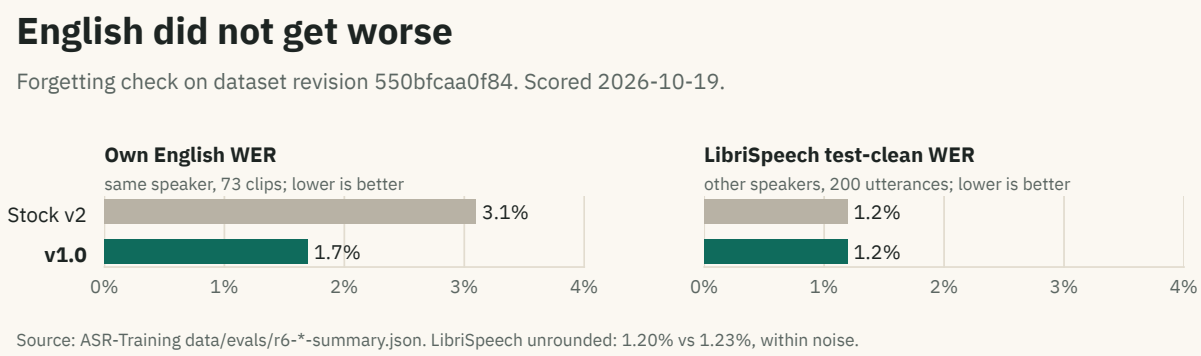

Figure 6. Forgetting check for Jachnun. Own English is the same speaker; LibriSpeech is other speakers.

Jachnun's own-English WER fell from 3.1% to 1.7%, and the public benchmark's English control from 2.7% to 2.2%. On LibriSpeech, other people's English, it scored 1.20% against stock's 1.23%: a noise-level margin, which is the point; it passed the project's forgetting gate. Sabich's own English (1.62%) and Hebrew-free English (1.23%) stay well below stock (2.81%, 1.79%). On LibriSpeech Sabich scored 1.30% against stock's 1.23%, a strict fail of the gate by about 3 words in 200 clips, which Daniel Rosehill accepted as noise.

What the results say

- **Hebrew words.** Both models more than double stock Parakeet's term recall on the public benchmark (73.3% and 73.8% against 32.6%) and lead every model scored, including Whisper Large V3 (45.9%) and the earlier Whisper fine-tune (54.1%). Jachnun's unseen-term recall (70.1%) is close to its overall recall, which suggests the gain is not only memorisation of training terms.
- **Code-switched WER** fell from 11.3% (stock) to 5.1% on the `550bfcaa` held-out split, and from 9.5% to 3.3% on the public benchmark.
- **Sabich against Jachnun.** The extra tech and task data cost nothing on code-switching (5.30% against 5.42% WER, 48.15% against 46.91% held-out recall, level on the public benchmark) and gained modestly on tech dictation (1.54% against 1.74%). That is why Sabich became the apps' recommended model on 2026-10-21. The gaps are small and come from one round of tests on one voice.
- **Against PoC-3**, Jachnun is better on every public-benchmark metric. On the held-out split, the gain on test terms seen in training (+9.4 points) is real; the gain on held-out terms (+2.1 points, about 6 of 293 hits) is within noise. More data taught the words it contained; it did not clearly improve guessing words it never heard.

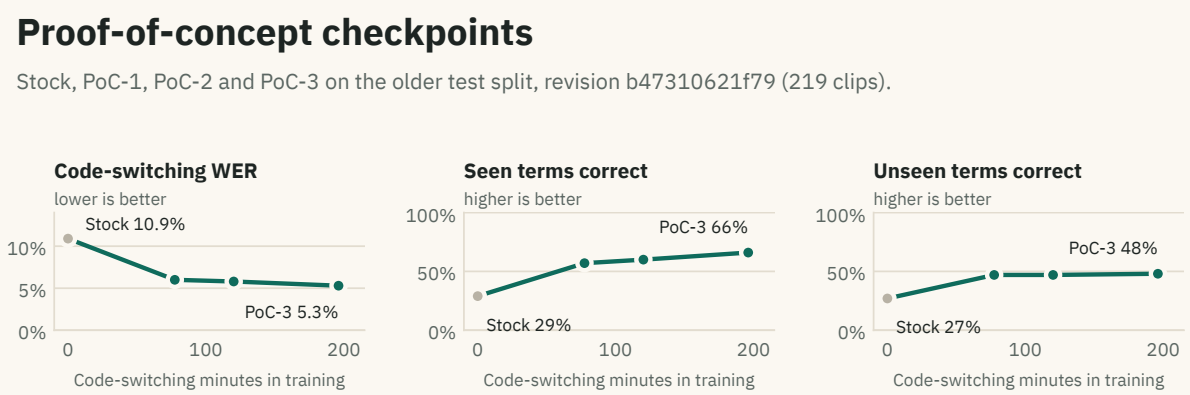

Figure 7. The proof-of-concept checkpoints on the older test split (revision `b47310621f79`, 219 clips). Not comparable with Figures 4 and 5.

9. In progress: Shakshuka, the streaming line

> **Not part of release 1.** Shakshuka is under construction. Its checkpoints are proofs of concept, it is not in the Hebrish apps, and its streaming accuracy has not been measured.

Shakshuka applies Jachnun's recordings and recipe to NVIDIA's cache-aware streaming model, `nvidia/nemotron-speech-streaming-en-0.6b`, so that text can appear while the speaker is still talking. Two checkpoints exist, both in `Hebrish/Nemotron-Speech-Streaming-EN-0.6B-English-Hebrew-Codeswitching` (access on request; DOI `10.57967/hf/11107`):

Checkpoint	Run	Released	Data
Shakshuka PoC-1	`r5-nemotron-en30`	2026-10-18	Trained beside PoC-5; in the repository history
Shakshuka PoC-2	`r6-nemotron-en30`	2026-10-19	Jachnun v1.0's data and recipe; best epoch 18, validation WER 6.3% (raw), 45.8 min

Scored **offline** (full context), Shakshuka PoC-2 trails Jachnun on every metric: 62.2% term recall and 4.7% WER on the public benchmark, 5.8% code-switched WER on the `550bfcaa` held-out split. It wrote none of the 8 number tokens in that test correctly. It has no forgetting verdict, because stock Nemotron was not re-scored on that revision. Next: a first retrain, Shakshuka v1.0, and an evaluation in streaming mode, which is what the line exists for.

10. Limitations

- **One speaker's voice.** Every clip, train and test, is the same speaker: Irish-born, English-speaking, living in Jerusalem. Only the LibriSpeech replay and test-clean are other speakers. The results rank models on this speaker and say nothing yet about other accents. More voices are the project's main need.
- **Read speech, mostly one phone.** Prompted sentences read aloud are cleaner than spontaneous dictation.

- **Small test sets.** 84 clean clips and 172 term occurrences on the public benchmark; a 2 to 3 point recall difference is within noise. The held-out splits are larger but have the same speaker.
- **Not every benchmark term is Hebrew.** 22 of the 139 clean-subset terms are English tech labels or product names (*JSON*, *grep*, *Proxmox*, ...), 26 of the 172 occurrences. Excluding them changes no ranks but lowers every model's recall by 2 to 6 points. A relabelled test set will be a new version.
- **A possible held-out leak is being checked.** Two held-out-list terms, *mandatim* and *Beit Safafa*, appear as training target terms at a later dataset revision (`7bfcea1d`). It is not yet confirmed how they were routed at `550bfcaa`. If they were trained on, their held-out hits are not unseen.
- **Held-out gains over PoC-3 are within noise** (Section 8). PoC-4 and PoC-5 were never benchmarked.
- **Quantized exports** were checked on five or ten clips each, not benchmarked.
- **Not evaluated:** a 30-epoch Sabich variant, `r7-all-e30` (validation WER 4.17%), exists but has not been scored on either test set.
- **The canonical spelling is a convention.** A model writing *tsaharon* for *tzaharon* is not wrong in an absolute sense; the benchmark reports lenient recall alongside.

11. Compute

Step	Compute
Training run `r6-en30` (Jachnun)	41.0 min on one L40S (Modal)
Training run `r7-all` (Sabich)	52.1 min on one L40S (Modal)
Benchmark round, 2026-10-19	Six parallel L40S jobs, 19:44–19:46 UTC, about 2.5 min wall each, under 15 GPU-minutes in all

The binding constraint is recording, not compute. Retraining and benchmarking a model this size takes about an hour of GPU time, so the project retrains on cumulative data whenever enough new audio has been recorded.

12. Links

- Website, models and benchmarks: hebrish.app/models · hebrish.app/benchmarks · datasets: hebrish.app/datasets
- Jachnun v1.0: Hebrish/Jachnun · Sabich v1.0: Hebrish/Sabich (access on request); every build at huggingface.co/Hebrish
- Shakshuka: Hebrish/Nemotron-Speech-Streaming-EN-0.6B-English-Hebrew-Codeswitching
- Proof-of-concept record: danielrosehill/Parakeet-TDT-0.6B-v2-English-Hebrew-Codeswitching (PoC-1 to PoC-5)
- Leaderboard: English-Hebrew-Codeswitching-ASR-Leaderboard · benchmark data: `Hebrish/English-Hebrew-Codeswitching-ASR-Benchmark`
- This paper: hebrish.app/whitepaper/hebrish-release-1-whitepaper.pdf · Hebrew: hebrish-release-1-whitepaper-he.pdf

Daniel Rosehill. This edition, 2026-10-21, reframes the earlier "Hebrish v1.0" white paper (2026-10-19, with an addendum on Sabich v1.0 of 2026-10-20) as model release 1, covering both models, and adds Section 9 on Shakshuka and Figure 5. No result changed. Text licensed **CC-BY-4.0**. The models inherit CC-BY-4.0 from `nvidia/parakeet-tdt-0.6b-v2`; LibriSpeech-PC, used for replay, is CC-BY-4.0; Shakshuka's base is under the NVIDIA Open Model License. All results are from one speaker's voice. Numbers are copied from the project's result files without rounding beyond their published precision.