How much capacity does Turkish inflection require? An empirical study of GRU encoder–decoder bottlenecks

Fred Mailhot
Dialpad, Inc.
fred.mailhot@dialpad.com

1 Introduction

Encoder–decoder neural networks with high-dimensional embeddings and hidden layers (typically d=300–500) are now the default tool for modeling morphophonological phenomena as sequence-to-sequence mappings, achieving high accuracy across languages and pattern types (cf. Kirov and Cotterell, 2018; Corkery et al., 2019; Mayer and Nelson, 2020; Prickett, 2021; Li et al., 2024, *inter alia*). For interpretability and cognitive-modeling purposes, this raises an obvious but under-examined question: how much of that capacity is necessary? Earlier connectionist work on similar problems succeeded with networks of single-digit hidden dimensionality (e.g. Rodd, 1997, for vowel harmony), raising the possibility that contemporary models may be substantially overparameterized, at least for some morphophonological tasks.

We address this question here for the case of Turkish morphological inflection by training very small GRU encoder–decoder models *without* attention, sweeping the dimensionality of an enforced bottleneck between the encoder and decoder. The attention-free design is deliberate: the bottleneck enforces a strict constraint on the information that propagates from encoder to decoder, so the question "how much capacity is needed?" becomes a measurable property of the data and task rather than of the architecture.

We report three findings: (i) sequence accuracy follows a sigmoid shape in bottleneck dimension, with a transition zone between h=16 and h=48, with performance near ceiling at h=64 (∼70k parameters), (ii) logistic regression probes on the bottleneck recover the harmony class [±back] of stem vowels at $\geq 95\%$ accuracy across *all* sizes, including sizes at which sequence accuracy is near zero, (iii) adding Bahdanau attention does not eliminate the bottleneck's role: sequence accuracy for attention without context transmission collapses to $\sim 1\%$, producing harmonically fluent but morphologically wrong outputs. Together, these results give a tight, replicable lower bound on the capacity required for this task in this architecture, and a clean picture of what the bottleneck encodes *before* it can support the full inflection task.

2 Methods

2.1 Data

We use the Turkish dataset from the CoNLL–SIGMORPHON 2018 Shared Task ($10k$ train, $1k$ dev). Inputs are *(lemma, feature-bundle)* pairs in Turkish orthography (e.g. *artık*, `N;GEN;PL`) and outputs are inflected surface forms (e.g. *artıkların*).

2.2 Model

Our model has a two-part encoder: a bidirectional GRU that encodes the input stem character sequence, and a pooled-embedding (d=8) MLP that encodes the morphosyntactic feature bundle. The stem encoder's final hidden state and the feature representation are concatenated and linearly projected (with *tanh* activation) to an h-dimensional *bottleneck context* vector. A unidirectional GRU decoder generates the output character-by-character, receiving the context vector concatenated to the character embedding at every timestep (teacher forcing during training; greedy decoding at inference). Stem encoder and decoder share (tied) d=8 character embeddings.[1]

2.3 Training

We train with AdamW: lr=10^{-3}, weight decay=10^{-3}, `ReduceLROnPlateau` scheduling, batch size=32, gradient clipping=1.0, and early stopping (patience=15 on validation loss).

[1] Character embedding dimension was swept from d=3 to d=32; performance exhibited a sharp transition at d=8, which we adopted as the fixed embedding size for the bottleneck sweep reported here.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 423–426

Models are trained on a Google Cloud VM with a single T4 GPU for a maximum of 400 epochs.

2.4 Bottleneck sweep

The bottleneck dimension h is swept across {8, 12, 16, 20, 24, 32, 48, 64, 96, 128} with five random seeds each (50 runs total). Model sizes range from $\sim$2.6–95k parameters.

2.5 Probing

To test whether the bottleneck encodes phonological structure, we extract context vectors from trained models and train logistic regression probes to predict the backness class ([±back]) of each stem's final vowel. We evaluate probe accuracy across all 10 hidden layer sizes. As controls—and to test whether the bottleneck encodes the features needed for Turkish's secondary rounding-harmony pattern—we train analogous probes for the rounding class ([±round]) and the binary height ([±high]) of the same vowel.

2.6 Attention ablation

To confirm the bottleneck is a genuine constraint rather than a data size or training artifact, we add Bahdanau attention and test three decoder configurations: (a) attention only (no context vector), (b) attention with context as decoder initialization (INIT_CTX), and (c) attention with context concatenated at every step (CONCAT_CTX). Each is swept over $h \in \{16, 20, 24, 32, 48\}$ with 5 random seeds.

3 Results

3.1 Sequence accuracy

Mean sequence accuracy (Figure 1) across hidden dimensions reveals a sigmoid-shaped curve, near-zero at h=8 (0.6%), rising steeply through the transition zone (h=16–48), and approaching ceiling by h=64 (92.0%). Variance peaks in the transition zone (h=24: σ=3.3) and is minimal at both extremes, consistent with a regime where small differences in initialization determine whether the model finds a good solution.

3.2 Harmony-aware edit distance

Exact-match sequence accuracy is a coarse signal: a model that fails to produce the right suffix may still produce a vowel of the correct harmonic class. We complement seq. acc. with a harmony-aware edit distance, in which vowel substitutions are weighted by phonological feature distance (backness and rounding) rather than treated as unit edits. This metric decreases monotonically in h from 6.5 at h=8 to 0.15 by h=96 — mirroring the sigmoid in seq. acc. but with no visible threshold. Even sub-ceiling models produce forms increasingly close to the target in feature space, consistent with the probing picture below.

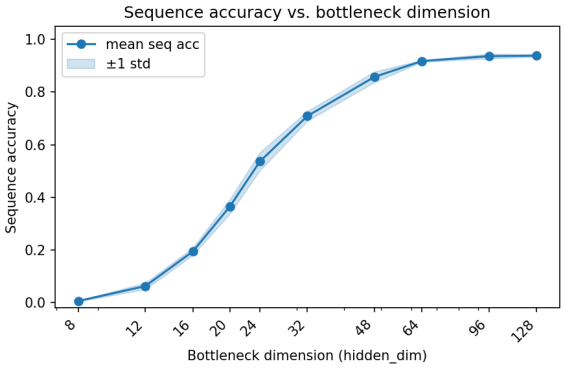

Figure 1: Sequence accuracy on dev set (mean ± std across 5 seeds) as a function of bottleneck dimension h.

h	[±back]	[±round]	[±high]	Seq. acc.
baseline	52.9	84.5	66.1	—
8	97.8	86.9	72.1	0.6
16	96.1	90.9	83.5	19.5
24	98.6	95.0	92.8	53.6
48	99.7	97.9	97.3	85.9
128	99.9	99.1	97.5	93.8

Table 1: Probe and sequence accuracy (%) vs. bottleneck dimension h. Probe columns predict [±back], [±round], and [±high] of the stem's final vowel; the *baseline* row gives majority-class accuracy on the training set (weighted by inflection frequency). [±back] is encoded well above baseline even at h=8, where inflection accuracy is near zero. [±round] and [±high] are both close to their respective baselines at low capacity and only diverge meaningfully at larger h.

3.3 Probing

Logistic-regression probes for [±back] achieve 96.1–99.9% accuracy across all bottleneck sizes (Table 1) – roughly 45 points above the 52.9% majority-class baseline. The dissociation at h=8 is striking: the model cannot produce correct outputs at all (0.6% seq. acc.), yet a linear classifier reads the stem's harmony class off the context vector essentially perfectly. *The bottleneck encodes what class the stem belongs to before it can encode enough to produce the right output.*

The dissociation does not extend to other vowel features. [±round] and [±high] have substantially higher majority-class baselines (84.5% and 66.1%

respectively), reflecting Turkish's phonotactic restriction on round vowels outside the first syllable and the lexical dominance of non-high vowels in stem-final position. Once baselines are accounted for, [±round] sits only 2.4 pp above baseline at h=8, and [±high] only 6.0 pp; both probes are essentially at majority-class performance at low capacity. By h=48 they rise to +13.4 pp and +31.2 pp above baseline respectively.[2]

One might worry that [±back] decoding is so successful at h=8 simply because Turkish orthography makes front/back vowel identity directly readable from individual characters—that the probe is recovering a surface graphemic property rather than harmonic structure. But Turkish orthography is transparent for *all three* vowel features we test: [±back], [±round], and [±high] are each fully determined by individual vowel characters. If trivial orthographic decoding were the explanation, all three should be equally recoverable above their respective baselines. They are not—only [±back] is.

The picture is therefore narrower and sharper than a generic "phonology before morphology" claim: the bottleneck does not encode vowel features in general at low capacity; it encodes the *one* binary distinction Turkish harmony is built on. [±back] is the feature the decoder must propagate to choose between front- and back-harmonic suffix allomorphs; encoding it is necessary, though not sufficient, for the full task. The features required only for the conditioned secondary rounding pattern emerge only with capacity.

3.4 Attention ablation

Adding attention *without* the context vector is catastrophic (Table 2): accuracy drops to ∼1% at all hidden dimensions. These models produce fluent, harmonically consistent Turkish—but with wrong suffixes, because the decoder never receives the morphosyntactic feature specification. Notably, harmony-aware edit distance for these models is *lower* than for the smallest baseline models (∼5.5 vs. 6.5 at h=8), confirming that they produce harmonically well-formed but morphologically incorrect output.

Restoring context access recovers performance. Initializing the decoder hidden state from the context vector (INIT_CTX) partially restores the capacity curve (31.6% at h=16, 94.7% at h=48), but with high variance at small sizes. Concatenating the context at every step (CONCAT_CTX) flattens the curve: even h=16 models reach 84.7%, and performance is near-ceiling by h=24 (92.5%).

Condition	h=16	h=20	h=24	h=32	h=48
No attention	19.5	36.6	53.6	70.9	85.9
Attn, no ctx	1.1	1.4	1.3	1.0	1.5
Attn+init_ctx	31.6	65.2	79.5	91.3	94.7
Attn+concat_ctx	84.7	90.3	92.5	94.5	95.8

Table 2: Sequence accuracy (%) across attention conditions. "No attention" is the baseline from Figure 1. All values are means over 5 seeds.

The contrast between INIT_CTX and CONCAT_CTX is informative. When the context is only available at initialization, it degrades over the course of generation—the decoder must maintain feature information in its recurrent state while simultaneously tracking the output sequence. When context is available at every step, this burden is removed, and the attention mechanism's ability to look up stem characters provides the remaining information needed for inflection.

4 Discussion

4.1 What capacity is "enough"?

The sequence-accuracy curve has a clear transition zone (h=16–48) and reaches ceiling by h≈64. We emphasize that this is a lower bound for *this architecture* on *this task and dataset*: a GRU encoder–decoder without attention, trained on the 10k-example CoNLL–SIGMORPHON 2018 Turkish split. We do not claim that 64 hidden units is the universal capacity floor for Turkish inflection—only that, under standard training, more capacity than this is unnecessary, and meaningfully less is insufficient.

4.2 A property of the data, not (just) the network

The dissociation between probe accuracy and task accuracy invites a learnability reading. A bottleneck of h=8 has too little capacity to support correct inflection, but enough to carry a binary distinction over the input; and the binary distinction the model finds first is exactly the one Turkish harmony is built on. We read this as a claim about the input distribution rather than about a particular network: in a corpus of Turkish word forms encoded as orthographic strings, the [±back] partition

[2] Probes use the same logistic-regression setup for all three features; only the target label changes. [±high] is binarised as high vs. mid/low.

of the lexicon is the single most informative feature for predicting suffix shape, and it is the easiest to recover from finite samples even under severe representational constraint. The relevant caveat is that orthography is not raw signal: Turkish orthography is near-phonemic, so character-level inputs already deliver vowel features almost without distortion. Whether the same pattern would hold for a model trained on speech or a less transparent script is a separate question.

4.3 Attention does not erase the bottleneck

The attention ablation shows that allowing the decoder to attend over stem characters is not a substitute for transmitting the morphosyntactic feature bundle. Attention-only models reach $\sim 1\%$ accuracy at every h tested, while producing harmonically well-formed Turkish (mean harmony-edit distance ≈ 5.5, vs. 6.5 for the smallest no-attention baseline). When the bottleneck is restored—even just as decoder initialization—accuracy rises with h in the familiar way. Reinjecting context at every step (CONCAT_CTX) flattens the capacity curve almost entirely. The bottleneck's load-bearing role is the morphosyntactic feature bundle, not stem phonology per se: phonology can be re-fetched via attention; the feature specification cannot.

5 Scope and limitations

Our results are limited along three axes. *Architecture*: we report numbers for GRU encoder–decoders only; LSTM, Transformer, and convolutional variants may have different capacity profiles. *Language*: Turkish is a typologically agglutinative language with transparent (orthographically marked) vowel harmony; we expect quantitatively different curves for fusional or polysynthetic systems, and for harmony systems with transparent or neutral vowels (e.g. Finnish). *Data scale*: the SIGMORPHON 2018 *high* split is small relative to modern training corpora; minimum-capacity thresholds may shift under different data regimes. We therefore present our central quantitative claims (transition zone, ceiling capacity) as architecture- and corpus-specific empirical lower bounds, not as universal capacity floors.

6 Future work

The most immediate extension is cross-linguistic: applying the same bottleneck sweep to Finnish (which has transparent vowels and a richer harmony inventory) would test whether the capacity threshold for transparent-vowel harmony is meaningfully higher than for Turkish's opaque system. More broadly, comparable sweeps on a fusional language (e.g. Russian) and a polysynthetic one (e.g. Yupik) would calibrate whether the "small models suffice" finding generalises beyond agglutinative morphology, or whether it depends on the comparatively local conditioning of Turkish inflection. Finally, the dissociation between [±back] encoding and [±round]/[±high] non-encoding we observe at low capacity is a candidate experimental probe for cognitive learnability claims; replicating it under inputs that are not orthographically near-phonemic (raw audio features, or non-transparent scripts) would test how much of the effect we attribute to the data is in fact an artefact of the input encoding.

References

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877, Florence, Italy. Association for Computational Linguistics.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting Pinker and Prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Jane Li, Kyle Rawlins, and Paul Smolensky. 2024. What representations do rnns learn and use from morpho-phonological processes? In *Society for Computation in Linguistics*, volume 7, page 289–290.

Connor Mayer and Max Nelson. 2020. Phonotactic learning with neural language models. In *Society for Computation in Linguistics*, volume 3, pages 291–301.

Brandon Prickett. 2021. *Learning Phonology with Sequence-to-Sequence Neural Networks*. Ph.D. thesis, University of Massachusetts Amherst.

Jennifer Rodd. 1997. Recurrent neural-network learning of phonological regularities in Turkish. In *CoNLL97: Computational Natural Language Learning*.