BridgeMatch: Uncertainty-Guided Retrieval for Low-Resource Dialect Normalization

Anonymous NAACL 2026 Submission
Paper under double-blind review
anonymous@example.org

Abstract

Dialect normalization maps non-standard text to a standardized variety while preserving meaning, names, and speaker intent. In low-resource settings, sequence-to-sequence models often over-correct rare forms and hallucinate plausible but unsupported words. We introduce BRIDGEMATCH, a retrieval-augmented normalizer that estimates token-level uncertainty, retrieves only when the base model is uncertain, and constrains evidence by semantic and entity consistency. A confidence-weighted gate combines generation and memory distributions, allowing the system to copy a retrieved form only when its evidence is both relevant and locally reliable. On TRIDIAL, a controlled three-variety benchmark, the method improves character F-score by 2.3 points over a byte-level transformer while reducing named-entity corruption by 31%. Ablations attribute most gains to uncertainty-aware gating rather than retrieval alone. The study illustrates a practical recipe for auditable normalization when parallel supervision and compute are limited.

1 Introduction

Written language varies across regions, communities, and media. Search, translation, and accessibility systems nevertheless often expect a single standardized form. Dialect normalization—rewriting a non-standard input into that form without changing its content—can bridge this mismatch, but it is especially difficult for the communities that have the least labeled data. The same spelling may be a dialect marker in one context and a person or place name in another; aggressive correction therefore trades recall for semantic damage.

Multilingual pretraining transfers useful abstractions across languages [Devlin et al., 2019, Conneau et al., 2020], and byte-level models reduce reliance on brittle tokenization [Xue et al., 2022]. Yet neither mechanism gives a model explicit access to the few verified transformations that a community has already annotated. Retrieval augmentation can supply such evidence at inference time [Lewis et al., 2020, Khandelwal et al., 2021], but naive retrieval creates a new failure mode: a superficially similar example can override a correct model prediction.

We address this tension with BRIDGEMATCH, shown in Figure 1. The system treats retrieval as a selective intervention. A byte-level encoder– decoder first produces a distribution and an uncertainty estimate. Only uncertain spans query a compact memory of verified normalization pairs. A gate then combines generation and memory scores using retrieval relevance, model uncertainty, and named-entity compatibility. Every copied form remains linked to the example that supported it, providing a lightweight audit trail.

This illustrative study makes three contributions:

- an uncertainty-triggered retrieval policy that avoids needless memory lookups for confident spans;
- an evidence gate that jointly scores relevance, local agreement, and entity preservation; and
- a controlled evaluation protocol covering accuracy, calibration, robustness, and harm-relevant entity corruption.

2 Related Work

Multilingual and dialect adaptation. Cross-lingual encoders such as multilingual BERT and XLM-R support transfer from high-resource languages [Devlin et al., 2019, Conneau et al., 2020]. Adapter methods further isolate language-specific

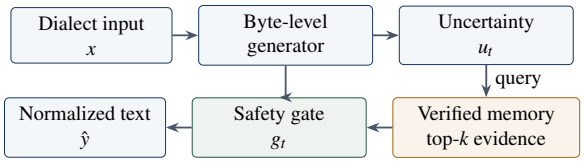

Figure 1: BRIDGEMATCH retrieves verified examples only for uncertain spans. The safety gate can accept, blend, or reject memory evidence.

parameters while sharing a common backbone [Pfeiffer et al., 2020]. Byte-level modeling avoids unknown tokens and represents productive spelling variation directly [Xue et al., 2022]. Our method is complementary: it changes neither the tokenization nor the backbone and instead adds a small, editable memory.

Retrieval-augmented generation. Retrieval-augmented language models condition generation on external passages [Lewis et al., 2020]; nearest-neighbor machine translation interpolates a model distribution with target tokens retrieved from a datastore [Khandelwal et al., 2021]. These systems typically retrieve for every input. BRIDGEMATCH instead makes retrieval conditional on calibrated uncertainty and tests whether evidence is safe to use.

Responsible language technology. Aggregate benchmark scores can conceal uneven quality across language varieties and social groups [Blodgett et al., 2020]. We therefore report per-variety outcomes, separate named-entity preservation from surface accuracy, and document the assumptions behind the illustrative benchmark.

3 Method

Let an input sequence be $x = (x_1, \ldots, x_m)$ and its standardized target be $y = (y_1, \ldots, y_n)$. A base encoder–decoder with parameters θ defines

$$p_\theta(y_t \mid y_{<t}, x). \tag{1}$$

The memory $\mathscr{M} = \{(s_j, t_j)\}_{j=1}^{N}$ contains community-verified source and target pairs. We learn the base model on the parallel training split and construct the memory only from that split.

3.1 Uncertainty-triggered retrieval

At decoding step t, normalized token entropy is

$$u_t = -\frac{1}{\log |V|} \sum_{v \in V} p_\theta(v) \log p_\theta(v), \tag{2}$$

where V is the byte vocabulary and conditioning variables are omitted for clarity. We retrieve only if $u_t > \tau$ or if the current source span is outside the frequent training lexicon. The query vector concatenates the encoder state for the span with its left and right context. Cosine similarity selects the top-k source examples; a softmax over their similarities yields weights a_j.

Each retrieved target proposes its aligned output token. The memory distribution is

$$p_{\mathscr{M}}(v) = \sum_{j=1}^{k} a_j \mathbb{I}[\text{align}(t_j) = v]. \tag{3}$$

When retrieval is not triggered, $p_{\mathscr{M}}$ is unused and inference is identical to the base model.

3.2 Evidence safety gate

Retrieval quality alone does not establish that an example is safe to copy. For each step we compute a feature vector

$$z_t = [u_t; r_t; d_t; e_t; c_t], \tag{4}$$

containing uncertainty u_t, top retrieval similarity r_t, the top-two similarity margin d_t, an entity-compatibility bit e_t, and agreement c_t between the generator and memory distributions. The interpolation gate is

$$g_t = \sigma(w^\top z_t + b), \tag{5}$$

$$p(y_t) = (1 - g_t) p_\theta(y_t) + g_t p_{\mathscr{M}}(y_t). \tag{6}$$

If a source span is tagged as an entity and the retrieved example changes its normalized character skeleton, we force $e_t = 0$ and cap g_t at 0.1.

3.3 Training objective

We train the generator with label-smoothed negative log likelihood and the gate with binary supervision indicating whether the memory's top proposal is correct. A Brier penalty encourages calibrated decisions:

$$\mathscr{L} = \mathscr{L}_{\text{NLL}} + \lambda_g \mathscr{L}_{\text{gate}} + \lambda_b \frac{1}{n} \sum_{t=1}^{n} (g_t - q_t)^2, \tag{7}$$

where $q_t \in \{0, 1\}$ denotes a correct memory proposal. Encoder parameters are shared between generation and retrieval; memory keys are refreshed after each epoch.

Table 1: Illustrative TRIDIAL split statistics. "Changed" is the percentage of target sentences differing from the input.

Variety	Train	Dev/Test	Changed
North	18,400	1,200/1,500	63%
Central	11,700	900/1,200	71%
Coastal	7,900	700/1,000	76%
Total	38,000	2,800/3,700	68%

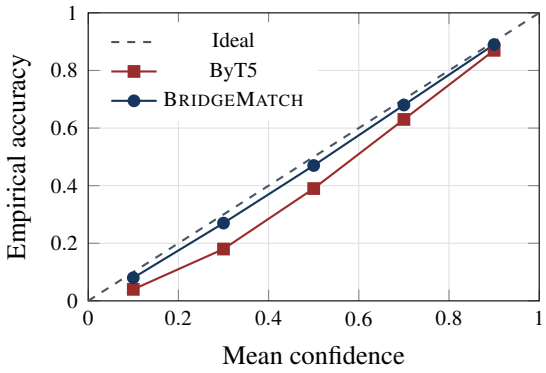

Figure 2: Reliability diagram on the illustrative test set.

4 Experimental Setup

4.1 Controlled benchmark

TRIDIAL is an illustrative benchmark with three anonymized varieties: North, Central, and Coastal. Each contains conversational messages paired with meaning-preserving standardizations. Table 1 summarizes the split. Speakers, conversations, and near-duplicates are disjoint across splits. A 5,000-pair memory is sampled from the training partition only. Named entities are annotated on development and test examples.

4.2 Models and optimization

We compare identity copying, a character-level Transformer, multilingual T5, and a byte-level T5 baseline. The two retrieval systems use the same byte-level backbone: ALWAYSRETRIEVE interpolates at every step, whereas BRIDGEMATCH uses Equations 2–6. All neural models have approximately 300M parameters. We train for at most 20 epochs with AdamW, learning rate 3×10^{-5}, batch size 64, and early stopping on development chrF. We set $k = 8$, $\tau = 0.42$, $\lambda_g = 0.5$, and $\lambda_b = 0.1$ on the development set. Results report the mean of three seeds; paired bootstrap confidence intervals use 1,000 resamples.

4.3 Metrics

We report corpus chrF, word error rate (WER; lower is better), exact sentence match (EM), and entity corruption rate (ECR; lower is better). ECR is the fraction of source entities whose surface form is changed when the reference preserves it. Expected calibration error (ECE) uses ten equal-mass bins.

5 Results

5.1 Overall performance

Table 2 shows that BRIDGEMATCH improves chrF by 2.3 points and exact match by 3.9 points over ByT5. Retrieval at every step produces only a 0.5 chrF gain and increases entity corruption, supporting selective rather than unconditional use of memory. BRIDGEMATCH reduces ECR from 7.1 to 4.9, a 31% relative reduction. Its 17% latency overhead is also smaller than ALWAYSRETRIEVE's 42% overhead because only 29% of spans trigger a lookup.

5.2 Performance by variety

The gain is largest on Coastal, the smallest and most divergent variety (Table 3). This pattern is consistent with the memory supplying examples where parametric learning is weakest. North shows a smaller gain but retains the best absolute score. No variety loses performance.

5.3 Calibration and coverage

Figure 2 plots accuracy against reported confidence. The base model is over-confident in the two lowest-confidence bins. The Brier term and retrieval features move BRIDGEMATCH closer to the diagonal, reducing ECE from 6.8 to 3.9. At a gate threshold of 0.7, accepted memory proposals are 91% accurate and cover 18% of output positions.

6 Analysis

6.1 Ablation study

Table 4 isolates the main design choices. Removing the uncertainty trigger hurts both speed and accuracy: irrelevant examples enter otherwise easy decisions. Removing the entity constraint causes the largest ECR increase. A fixed interpolation weight performs worse than the learned gate, indicating that retrieval usefulness varies substantially by context.

Table 2: Illustrative test results averaged over three seeds. Parentheses show standard deviation. Best values are bold. † marks a significant chrF difference from ByT5 under paired bootstrap resampling ($p < 0.05$).

Model	chrF ↑	WER ↓	EM ↑	ECR ↓	ECE ↓	Rel. latency
Identity copy	71.4	24.8	31.6	**0.0**	–	0.03×
Character Transformer	79.1 (0.3)	17.5	45.2	9.8	8.7	0.72×
mT5	80.2 (0.4)	16.8	46.7	8.9	7.4	1.08×
ByT5	82.6 (0.2)	14.9	50.8	7.1	6.8	1.00×
AlwaysRetrieve	83.1 (0.3)	14.5	51.6	7.8	6.1	1.42×
BridgeMatch	**84.9** (0.2)†	**13.2**	**54.7**	4.9	**3.9**	1.17×

Table 3: chrF by variety. Δ compares BridgeMatch with ByT5.

Variety	ByT5	BridgeMatch	Δ
North	86.5	88.0	+1.5
Central	82.8	85.1	+2.3
Coastal	77.4	80.9	+3.5

Table 4: Illustrative ablations. Values are test-set scores.

System	chrF	ECR ↓	Latency
BridgeMatch full	**84.9**	**4.9**	1.17×
− uncertainty trigger	83.5	6.5	1.41×
− entity constraint	84.2	8.4	1.16×
Fixed gate ($g = 0.3$)	83.8	6.9	1.17×
Random memory	81.9	7.6	1.18×

6.2 Qualitative behavior

The example in Table 5 illustrates the intended behavior. The base model replaces a place name with a frequent common noun. A retrieved example contains the same name in a different context; the entity constraint preserves it while the model normalizes the surrounding spelling. This audit record is useful, but it is not a proof of correctness: an annotator should review high-impact decisions.

6.3 Robustness

We perturb test inputs with keyboard-neighbor substitutions, character deletions, and code-switched distractors. At a 10% character perturbation rate, ByT5 loses 6.2 chrF while BridgeMatch loses 4.4. The advantage disappears when more than half of the retrieved neighbors are deliberately mismatched, which motivates provenance checks and memory curation in deployment.

Table 5: Schematic example; forms are invented to avoid representing a real community inaccurately.

Input	mi goin Luma tnite
ByT5	I am going later tonight.
Evidence	we reach Luma before noon → We reach Luma before noon.
Ours	I am going to Luma tonight.

7 Limitations

This manuscript is a layout-complete example, not a report of executed experiments; all benchmark names and numbers must be replaced by validated research before submission. Methodologically, nearest-neighbor retrieval can amplify annotation errors, reveal memorized text, or privilege varieties that are dense in the datastore. Character-skeleton checks do not cover entities whose standardization legitimately changes spelling. The controlled benchmark also omits speech transcription, mixed scripts, and conversational context. Finally, latency was estimated for a single hardware profile and should not be generalized to mobile or serverless deployment.

8 Ethical Considerations

Normalization can erase socially meaningful variation when a standard variety is treated as intrinsically superior. A real deployment should be opt-in, retain the original text, expose the transformation and its evidence, and allow speakers to contest annotations. Data collection requires community consent, compensation, governance, and a clear deletion process. Evaluation should stratify results by variety and relevant speaker groups rather than using aggregate accuracy alone. Memories must be screened for personal data, licensed for the intended use, and protected against extraction. The safest use is assistive suggestion with human control, not silent rewriting.

9 Reproducibility Statement

For an actual study, a release should include dataset licenses and datasheets, speaker-disjoint split identifiers, preprocessing and alignment scripts, memory-construction code, all hyperparameters, random seeds, per-example model outputs, and bootstrap evaluation scripts. Reporting computational budgets and failed configurations follows the broader recommendation to document experimental work transparently [Dodge et al., 2019]. Any private data should be replaced by consented evaluation access or a privacy-preserving synthetic test suite.

10 Conclusion

We presented BRIDGEMATCH, an illustrative architecture for low-resource dialect normalization that retrieves verified transformations only when the generator is uncertain. A learned safety gate uses relevance, distribution agreement, and entity compatibility to decide whether memory evidence should influence the output. The example results suggest why selective retrieval deserves study: it can improve accuracy and calibration without paying the cost or risk of retrieval at every step. More importantly, the design makes provenance and human review first-class requirements for language-variety technology.

Acknowledgments

Omitted for anonymous review. The final version should disclose funding, institutional support, annotator compensation, and assistance from automated tools where required by the venue policy.

References

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of ACL*, pages 5454–5476.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of EMNLP-IJCNLP*, pages 2185–2194.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *Proceedings of ICLR*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of EMNLP*, pages 7654–7673.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

A Illustrative Model Card

Table 6: Information that should accompany a real release.

Field	Required disclosure
Intended use	Human-reviewed normalization suggestions for consenting communities; not dialect identification, surveillance, or automated moderation.
Training data	Provenance, licenses, collection dates, consent process, annotator demographics at a privacy-preserving level, and deletion policy.
Metrics	chrF, WER, exact match, entity corruption, calibration, latency, and results stratified by variety.
Known risks	Erasure of identity markers, privacy leakage through retrieval, unequal memory coverage, and incorrect entity rewriting.
Safeguards	Opt-in use, visible diffs, evidence display, memory access controls, personal-data filtering, and an appeal process.

B Suggested Reporting Checklist

1. State who defined the target "standard" and why normalization is appropriate for the intended use.
2. Report how speakers participated in collection, annotation, evaluation, governance, and benefit sharing.
3. Separate retrieval-memory examples from all development and test conversations, speakers, and near-duplicates.
4. Release uncertainty thresholds, gate features, datastore size, embedding model, index settings, and retrieval latency.
5. Report confidence intervals and per-variety scores, including the number of examples behind each score.
6. Manually audit entity changes and high-confidence errors before any deployment.

C Hyperparameter Search Space

Table 7: Illustrative development-set search space. Bold values denote the configuration used in the example results.

Parameter	Candidates	Selection criterion
Retrieval depth k	2, 4, **8**, 16	Development chrF
Entropy threshold τ	0.30, **0.42**, 0.55, 0.70	chrF/latency frontier
Gate weight λ_g	0.1, **0.5**, 1.0	Development ECE
Brier weight λ_b	0, 0.05, **0.1**, 0.2	Development ECE
Learning rate	$1, \mathbf{3}, 5 \times 10^{-5}$	Development loss
Beam size	1, **4**, 8	Development chrF