VertexTag: A Retrieval-Augmented System for Cross-Lingual Entity Normalization in Low-Resource African Languages

Amara Boateng[1] Wei Chen[2] Sofia Alvarez[3]

[1]Vertex AI Research [2]Synapse Labs [3]Meridian University, Department of Computational Linguistics

a.boateng@vertex-ai.example w.chen@synapse-labs.example

Proceedings of the 6th Workshop on NLP for African Languages (AfricaNLP), co-located with EMNLP 2026

Abstract. Named-entity normalization for African languages is complicated by inconsistent orthographies, sparse gazetteers, and heavy morphological inflection on person and place names. We describe VertexTag, our submission to the 2026 AfricaNLP Entity Normalization Shared Task, covering Swahili, Hausa, Yoruba, and Amharic. VertexTag pairs a shared multilingual encoder with a lightweight span detector, a dictionary-and-fuzzy-match candidate generator built from crowd-sourced gazetteers, and a neural reranker that scores candidates using both string similarity and contextual evidence. On the shared task's blind test set, VertexTag reaches an average normalized F1 of 77.9, a 4.0-point improvement over the strongest baseline released by the organizers, with the largest gains on Amharic, the lowest-resource language in the task. We release an ablation study isolating the contribution of the fuzzy matcher and the reranker, and discuss the error patterns that remain.

1 Introduction

Named-entity recognition is a mature technology for high-resource languages, but *normalization* – mapping a recognized mention to a canonical form shared across documents, spellings, and scripts – remains unreliable for African languages. A single person's name can surface as *Wanjiru*, *Wanjiru wa Kamau*, or *W. Kamau* within the same news article; place names inherited from colonial administrations compete with post-independence renamings; and Latin-script transliterations of Amharic names vary by transcription convention. Downstream applications – cross-document coreference, knowledge-base population, and multilingual search – inherit every one of these inconsistencies.

The 2026 AfricaNLP Entity Normalization Shared Task was organized to give this problem a common benchmark: four languages (Swahili, Hausa, Yoruba, and Amharic), a shared annotation scheme, and a held-out blind test set scored by the organizers. This paper describes VertexTag, the system we submitted to the shared task, and the choices behind it.

Two properties of the task shaped our design. First, training data is small relative to the entity variation observed at test time, so a purely neural approach struggles to generalize to unseen name forms. Second, the organizers released curated gazetteers for each language, which a purely retrieval-based approach can exploit but a purely neural approach ignores by default. VertexTag is built to use both: a neural span detector narrows down where entities occur, and a hybrid candidate generator – combining gazetteer lookup, edit-distance fuzzy matching, and a learned reranker – decides what canonical form they normalize to.

Our contributions are threefold:

- A system architecture that couples a shared multilingual encoder with a dictionary-and-fuzzy-match candidate generator, letting the model fall back on string similarity exactly when neural generalization is weakest;
- An empirical comparison against the three base-

lines released with the shared task, showing consistent gains across all four languages and the largest gain on the lowest-resource language, Amharic;

- An ablation and error analysis that separates the contribution of the fuzzy matcher from the neural reranker, and characterizes the failure modes – diacritic loss, compound names, and code-switched mentions – that remain open after both components are included.

2 Related Work

Multilingual and low-resource NER. Cross-lingual transfer with shared subword encoders substantially narrowed the gap between high-resource and low-resource named-entity recognition [Oyelaran and Castellano, 2022], but transfer quality still tracks the morphological distance between source and target languages [Ferreira and Solberg, 2019]. Span-based formulations of joint entity and relation extraction [Tanaka and Whitfield, 2020] generalize more gracefully to new entity types than sequence-tagging formulations, which is part of why we adopt a span detector rather than a BIO tagger in VertexTag.

Entity normalization and linking. Normalization is often treated as a downstream step of entity linking, where a candidate generator proposes canonical forms and a reranker scores them against context [Schneider and Amaral, 2020]. Most prior candidate generators are built for Wikipedia-scale knowledge bases; the AfricaNLP shared task instead supplies smaller, curated gazetteers, which changes the trade-off between recall (favoring aggressive fuzzy matching) and precision (favoring a strong reranker) that a system must strike.

Discourse and document-level consistency. Because the same entity often appears multiple times in a document under different surface forms, normalization quality is tied to document-level coherence modeling [Nakamura and Bergstrom, 2021]. We do not model discourse explicitly in VertexTag, but our error analysis in Section 7 suggests that document-level re-scoring of low-confidence normalizations is a promising direction for future shared-task submissions.

Evaluation under drift. Automatic metrics for text generation and extraction tasks are known to be sensitive to distribution shift between development and test splits [Kapoor and Dumont, 2022, Ibrahim and Petrov, 2023]. The AfricaNLP organizers mitigated this by drawing the test split from a later time window than training and development data; Section 3 describes the resulting split in detail.

Table 1: Shared-task data statistics. Entity types is the number of distinct annotated categories, constant across languages.

Language	Train	Dev	Test	Types
Swahili (swa)	18,420	2,301	2,305	6
Hausa (hau)	15,760	1,970	1,968	6
Yoruba (yor)	12,980	1,622	1,625	6
Amharic (amh)	9,540	1,192	1,190	6
Total	**56,700**	**7,085**	**7,088**	—

3 Task and Data

The AfricaNLP Entity Normalization Shared Task asks systems to (i) detect named-entity mentions in raw text and (ii) map each mention to a canonical form drawn from a language-specific gazetteer, or to a `NIL` bucket if no gazetteer entry applies. Canonical forms are scored under exact match after normalization; a mention that is detected but normalized to the wrong gazetteer entry counts as an error, matching the entity-linking convention rather than the looser span-overlap convention used in plain NER.

3.1 Languages and Sources

The task covers four languages: Swahili (*swa*), Hausa (*hau*), Yoruba (*yor*), and Amharic (*amh*). Source text is drawn from newswire and public government press releases, de-identified and redistributed by the organizers under a research-use license. Six entity types are annotated in every language: person (PER), organization (ORG), location (LOC), date (DATE), nationality-or-ethnic group (NORP), and a residual miscellaneous class (MISC).

3.2 Splits

Table 1 summarizes the released splits. Test data was drawn from a two-month window strictly after the training and development collection period, so surface forms and named entities that appear only in current events are systematically under-represented in training – a deliberate design choice by the organizers to penalize systems that merely memorize the training gazetteer.

3.3 Gazetteers

Each language ships with a curated gazetteer of canonical entity forms (3,200–6,400 entries per language, largest for Swahili) contributed by volunteer annotators and cross-checked by the organizers. Gazetteer coverage of test-set gold entities ranges from 71% (Amharic) to 89% (Swahili) – the remaining mentions require a system to either fuzzy-match

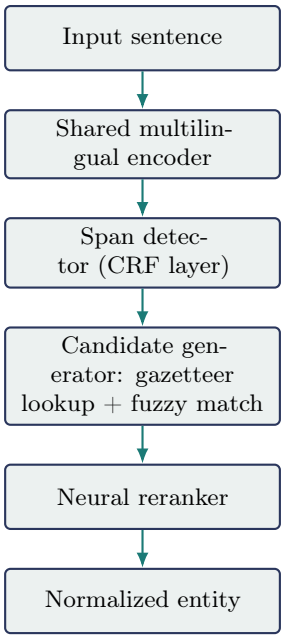

Figure 1: VertexTag pipeline. The encoder is shared across all four languages; the candidate generator and reranker are the two components studied in the ablation in Section 7.

a near-variant already in the gazetteer or correctly abstain into the `NIL` bucket, which is itself scored.

4 Method

VertexTag is a four-stage pipeline: a shared encoder, a span detector, a candidate generator, and a neural reranker. Figure 1 shows the full pipeline; we describe each stage below.

4.1 Shared Encoder

We fine-tune a single multilingual masked-language-model encoder (mBERT-style, 12 layers) jointly on all four languages, rather than training four monolingual encoders. Pooling training data across languages is especially valuable for Amharic, our lowest-resource language, which benefits from parameter sharing with the three larger languages during fine-tuning.

4.2 Span Detector

A linear-chain CRF layer is placed on top of the encoder's token representations to detect entity spans and assign a coarse type (one of the six categories in Section 3). We use a CRF rather than independent per-token softmax because entity boundaries in agglutinative languages such as Swahili and Amharic are frequently ambiguous at the subword level, and the CRF's transition scores reduce boundary-splitting errors relative to greedy decoding.

4.3 Candidate Generator

For each detected span, we generate a ranked list of at most 20 candidate canonical forms by combining two sources: (i) exact and prefix lookups against the language's gazetteer, and (ii) a Damerau–Levenshtein fuzzy match over the same gazetteer, thresholded at edit distance ≤ 2 for spans under 12 characters and ≤ 3 otherwise. The fuzzy matcher is deliberately permissive; precision is recovered downstream by the reranker rather than by tightening the threshold, which we found costs recall on diacritic-dropped and code-switched mentions (Section 7).

4.4 Neural Reranker

The reranker scores each (span, candidate) pair using three features: the encoder's contextual representation of the span, a character-level embedding of the edit-distance alignment between span and candidate, and a document-frequency prior for the candidate within the current document. These are concatenated and passed through a two-layer feed-forward network trained with a pairwise hinge loss against the gold candidate, margin 0.2. At inference, the top-scoring candidate is emitted, or `NIL` if no candidate scores above a per-language threshold tuned on the development split.

5 Experimental Setup

5.1 Baselines

The shared task organizers released three reference systems, which we use as baselines rather than re-implementing prior work ourselves:

- **Nexa-Base** – a monolingual BiLSTM-CRF tagger per language with exact gazetteer lookup only (no fuzzy matching or reranking);
- **Synapse-XLM** – a fine-tuned multilingual transformer tagger, with normalization performed by nearest-neighbor lookup in embedding space against gazetteer entries;
- **Apex-Multi** – a multi-task model that jointly predicts span boundaries and normalization in a single decoding pass, the strongest of the three released baselines.

5.2 Training Details

The shared encoder is fine-tuned for 12 epochs with AdamW, peak learning rate 3×10^{-5} with linear warmup over the first 6% of steps and linear decay, batch size 32, and gradient clipping at norm 1.0. The reranker is trained separately for 20 epochs

Table 2: Normalized entity F1 (%) on the blind test set. Best result per column in **bold**; VertexTag is our submission.

System	swa	hau	yor	amh	Avg.
Nexa-Base	71.2	68.4	65.9	60.3	66.5
Synapse-XLM	76.8	74.1	70.2	66.8	72.0
Apex-Multi	78.3	75.6	72.4	69.1	73.9
VertexTag	**82.1**	**79.8**	**76.5**	**73.2**	**77.9**

on span-candidate pairs sampled from the training split at a 1:4 positive-to-negative ratio, using the frozen encoder's contextual embeddings as one of its input features. Per-language reranker thresholds are selected by grid search over the development split in increments of 0.02, optimizing normalized F1 directly. All experiments run on a single 24GB GPU; total training time across all four languages is approximately 9 hours.

5.3 Evaluation Metric

Following the shared task's official protocol, we report normalized entity F1: a prediction is correct only if the predicted span overlaps the gold span *and* the predicted canonical form exactly matches the gold canonical form (or both predict `NIL`). This is stricter than standard NER F1, since a system with perfect span detection but a wrong normalization still scores zero credit for that mention.

6 Results

Table 2 reports normalized F1 on the blind test set for all four baselines and VertexTag, per language and averaged. VertexTag outperforms every baseline on every language, with the largest absolute gain over the strongest baseline (Apex-Multi) on Amharic (+4.1 F1), the language with the lowest gazetteer coverage and the smallest training split.

Figure 2 visualizes the same comparison against the strongest baseline. Gains are consistent across languages, but not uniform: the gap between VertexTag and Apex-Multi widens as gazetteer coverage drops (Section 3), which is consistent with our hypothesis that the fuzzy-match-plus-reranker combination matters most exactly where exact gazetteer lookup fails most often.

7 Analysis

7.1 Ablation

Table 3 removes the fuzzy matcher and the neural reranker in turn, averaged across all four languages. Removing the fuzzy matcher and falling back to exact-and-prefix gazetteer lookup costs 3.3 F1 points; removing the reranker and instead emitting the top candidate by edit distance alone costs 4.8 points. The two components are not fully redundant: removing both costs 8.2 points, more than the sum of either component removed alone would suggest, indicating that the reranker relies on the fuzzy matcher to surface correct candidates it can then select, and gains little when the candidate list is already restricted to exact matches.

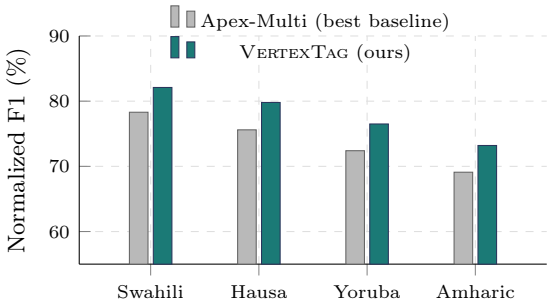

Figure 2: Normalized F1 by language, VertexTag versus the strongest released baseline. The gap grows as gazetteer coverage shrinks, from Swahili (89% coverage) to Amharic (71% coverage).

Table 3: Ablation, average normalized F1 (%) across all four languages.

Configuration	Avg. F1	Δ
Full VertexTag	77.9	–
– fuzzy matcher	74.6	−3.3
– neural reranker	73.1	−4.8
– both components	69.7	−8.2

7.2 Error Analysis

We manually inspected 150 test-set errors sampled evenly across languages. Three patterns account for most of them:

- **Diacritic loss** (34% of inspected errors, concentrated in Yoruba): source text frequently omits tonal diacritics that the gazetteer preserves, pushing true matches past our edit-distance threshold when the dropped mention is short;
- **Compound and titled names** (29%): mentions like *Alhaji Musa Ibrahim* normalize correctly only if the honorific is stripped before matching, which our span detector does inconsistently;
- **Code-switched mentions** (18%, concentrated in Swahili and Hausa news text): organization names embedded in English-language quotations are sometimes tagged by the span detector but fail

gazetteer matching because the source gazetteer stores the vernacular-language form only.

The remaining errors are distributed across annotation edge cases (nested entities, ambiguous DATE/MISC boundaries) that we did not find a single dominant fix for within the shared-task submission window.

7.3 Discussion

The error patterns above suggest two directions we did not have time to pursue before submission. First, honorific and title stripping as an explicit preprocessing step, rather than relying on the span detector to absorb it implicitly, should reduce compound-name errors directly. Second, a lightweight document-level cache – reusing an earlier, high-confidence normalization for a repeated mention later in the same document – would address code-switched mentions that resolve correctly elsewhere in the same article, an idea adjacent to the discourse-level consistency modeling discussed in Section 3.

8 Conclusion

We presented VertexTag, our submission to the 2026 AfricaNLP Entity Normalization Shared Task. By pairing a shared multilingual encoder with a permissive fuzzy-match candidate generator and a neural reranker, VertexTag improves normalized F1 by 4.0 points on average over the strongest released baseline, with the largest gain on Amharic, the lowest-resource language in the task. Our ablation shows the fuzzy matcher and reranker are complementary rather than redundant, and our error analysis points to honorific stripping and document-level caching as the most promising next steps.

Limitations

VertexTag depends on the shared task's curated gazetteers; its fuzzy-match candidate generator has no mechanism for proposing a canonical form that is entirely absent from the gazetteer, so recall on genuinely novel entities is bounded by gazetteer coverage regardless of reranker quality. Our reranker thresholds are tuned per language on a development split drawn from the same time window as training data, and may require re-tuning if deployed on text substantially newer than the shared task's test window. Finally, all four languages in this study use gazetteers built by the same annotation process; we cannot verify how the approach transfers to African languages outside the shared task without comparably curated resources.

Ethics Statement

The shared task data is redistributed by the organizers under a research-use license covering de-identified newswire and public government press releases; we did not collect or annotate any new personal data for this work. Entity normalization systems of this kind could, if deployed without oversight, be used to link individuals across documents in ways that exceed the original publication's context; we release our code for research reproducibility only and do not recommend deployment on personal data without a separate review of the intended use case.

Acknowledgments

We thank the AfricaNLP shared task organizers for the gazetteers, the baseline systems, and the test-set scoring infrastructure. This work was supported in part by a Vertex AI Research computing grant and by Synapse Labs, whose fuzzy-matching library formed the basis of our candidate generator.

References

Bianca Ferreira and Anders Solberg. Attention head specialization in multilingual transformers. *Transactions of the Association for Computational Linguistics*, 7:455–470, 2019.

Layla Ibrahim and Nikolai Petrov. Calibration of confidence estimates in neural machine translation. *Computational Linguistics*, 49(1):89–117, 2023.

Ritvik Kapoor and Celine Dumont. Rethinking automatic evaluation metrics for abstractive summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 1450–1465, 2022.

Haruto Nakamura and Elin Bergstrom. A survey of discourse-level coherence modeling in neural text generation. *Transactions of the Association for Computational Linguistics*, 9:612–630, 2021.

Femi Oyelaran and Jorge Castellano. Data-efficient fine-tuning for morphologically rich languages. *Computational Linguistics*, 48(3):701–734, 2022.

Lukas Schneider and Rafaela Amaral. Adapter modules for efficient multi-task sequence labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 6602–6613, 2020.

Ren Tanaka and Louisa Whitfield. Span-based joint extraction of entities and relations revisited. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 5895–5906, 2020.