Justin “Aurelio” Fernandez Sanchez

Prof. Matthew Malone

LNG 3430

12/16/2025

Multilingual POS Tagging Evaluation: Examining NLP Performance Disparities Across High-Resource and Low-Resource Languages

1. Abstract

This project investigates performance disparities in part-of-speech (POS) tagging across languages with different resource availability levels. Using Stanza, a neural network-based NLP toolkit trained on Universal Dependencies corpora, I evaluated POS tagging accuracy on four language varieties: English (high-resource), Standard Spanish (high-resource), Yoruba (low-resource), and Dominican Spanish (dialectal variety, low-resource). Results reveal a 78.2 percentage point accuracy gap between English’s 96.5% and Yoruba’s 18.3%, with Dominican Spanish achieving only 73.4% accuracy compared to Standard Spanish's 92.9%. Qualitative error analysis shows that models trained on standard varieties systematically fail on dialectal features including phonological contractions, code-switching, and non-standard lexical items. A controlled experiment equalizing corpus sizes across languages confirms that high-resource languages perform better with less data, indicating that infrastructure bias is embedded at the architectural level rather than merely reflecting training data availability. These findings quantify the systematic exclusion of low-resource languages and dialectal varieties from current NLP infrastructure.

2. Introduction

Natural language processing technologies have become ubiquitous in our digitized world, powering applications like machine translation, sentiment analysis, voice assistants, accessibility tools, and many more. However, the performance of these technologies varies dramatically across the world's ~7,000 languages. While state-of-the-art NLP models achieve near-human performance on high-resource languages like English and Mandarin Chinese, speakers of low-resource languages often encounter systems that fail to recognize basic grammatical structures in their native tongues.

This disparity extends beyond the level of language to dialectal variation within languages. Speakers of non-standard varieties frequently experience degraded performance even when using technologies seemingly designed for their "language." For instance, an English speaker using African American Vernacular English (AAVE), a Spanish speaker using the Boricua dialect, or an Arabic speaker using the Palestinian dialect may all encounter NLP systems that treat their native and completely natural speech patterns as errors that "mistakenly" deviate from the established "standard," rather than valid linguistic variation.

As someone of both Dominican heritage and Nigerian ancestry, I have personal investment in understanding how NLP technologies serve (or fail to serve) speakers of Dominican Spanish and Yoruba. My family speaks Dominican Spanish at home, characterized by features like phonological reduction (*ta* for *está*, *pa'* for *para*), distinctive lexical items (*vaina*, *tiguere*, *enratiao*), and extensive code-switching with English. Meanwhile, my study of Yoruba linguistic structures in my Spring 2025 class, LNG 339: African Languages with Matthew Malone, has

made me aware of its 48 million L1 speakers and rich grammatical system. However, I have consistently observed its absence from mainstream NLP tools and applications during my study of the language.

This project emerged from a fundamental question: do language technologies work equitably for speakers of these low-resource languages and varieties? To answer this question, I conducted a systematic evaluation of POS tagging accuracy across three languages and a dialect representing different points on the resource availability spectrum. POS tagging serves as an ideal test case because it represents a foundational NLP task that underlies virtually all widespread NLP applications, from parsing to machine translation to information extraction.

The research proceeds in two phases. First, I evaluate pretrained models as they exist in real-world deployment, measuring the actual performance that users of these languages would experience in day-to-day use. Second, I conduct a controlled experiment that equalizes corpus sizes across languages, isolating the effect of linguistic properties and model architecture from simple data volume. Throughout the analysis, I pay particular attention to the types of errors models make, asking not just how often they fail, but on what linguistic features they systematically struggle with.

3. Previous Research

3.1 Low-Resource NLP and Language Endangerment

The inequity low-resource languages face in NLP has received increasing attention in recent years, particularly as researchers have recognized that advances in neural methods have

disproportionately benefited high-resource languages (Joshi et al., 2020). Low-resource languages are typically defined as those lacking sufficient digital text data, annotated corpora, or pretrained models to support standard supervised machine learning approaches. However, this very definition obscures an important reality: many so-called “low-resource languages” have substantial speaker populations and established written traditions but are systematically excluded from data collection and model development pipelines.

Yoruba is a prime example of this exact paradox. With approximately 48 million L1 speakers and 2 million L2 speakers (SIL International, 2025), primarily in Nigeria and the West African diaspora, Yoruba ranks among Africa's most widely spoken native languages. (Ishola et al., 2020) It has standardized orthography, substantial written literature, and a stable presence in education, media, and government. However, despite this, Yoruba appears in few pretrained NLP models and limited digital corpora compared to European languages like say Finnish or Estonian. These languages possess far fewer speakers yet are spoken in wealthier regions with greater technological infrastructure, and have better access to NLP labs which disproportionately favor European languages.

Recent efforts have sought to address these disparities through initiatives like Masakhane, a grassroots organization focused on African language NLP (Orife et al., 2020), and the creation of multilingual benchmarks like XTREME (Hu et al., 2020). However, these valiant efforts face substantial challenges including but not limited to: limited funding, lack of standardized evaluation protocols, and the fundamental architectural biases of models trained primarily on English and other Indo-European languages.

3.2 Dialectal Variation and Standard Language Ideology

While low-resource languages face both challenges and exclusion from NLP infrastructure, speakers of non-standard dialects encounter a different but related problem: their "language" is recognized but systematically marked as "deviant" or "incorrect." For example, Blodgett et al. (2016) demonstrated that Twitter language identification systems misclassified AAVE tweets at substantially higher rates than those in General American English, with these errors correlating with features like habitual *be* and copula deletion that are grammatically regular in AAVE. Similarly, Jurgens et al. (2017) showed that sentiment analysis systems trained on standard varieties performed significantly worse on dialectal text, with errors particularly concentrated on code-switching and orthographic variation.

These patterns reflect what linguists call "standard language ideology" (Milroy, 2001), the belief that one variety of a language is inherently more correct, logical, or sophisticated than the rest. In NLP contexts, this exact ideology manifests in training data collection that privileges formal written registers, annotation guidelines that enforce prescriptive grammatical rules, and evaluation metrics that unjustly penalize variation from standard forms. The result of this bias is technological infrastructure that not just encodes, but enshrines social hierarchies regarding which speakers' language counts as "valid" and "correct."

Caribbean Spanish varieties, including Dominican Spanish, exhibit precisely the types of features that challenge standard-trained models. Phonological processes like syllable-final /s/ deletion and liquid consonant neutralization produce surface forms that differ substantially from standard Spanish orthography. Alongside this, extensive English contact, particularly in urban

centers and diaspora communities, results in frequent code-switching and lexical borrowing. Furthermore, unique lexical items emerge from West African, Indigenous, and local innovation (Alba, 2004). Despite these well-recorded linguistic phenomena, no substantial annotated corpora of Dominican Spanish exist, and speakers of Caribbean varieties report consistent errors in NLP applications like autocorrect, voice recognition, and machine translation.

3.3 Universal Dependencies and Cross-Linguistic Evaluation

The Universal Dependencies (UD) project (Nivre et al., 2016) provides a crucial infrastructure for cross-linguistic NLP research by standardizing annotation schemes across languages. UD defines 17 universal POS tags (e.g., NOUN, VERB, ADJ, ADP) and a consistent convention of dependency grammar, enabling direct comparison of model performance across typologically diverse languages. As of 2025, UD includes 339 treebanks in 186 languages, ranging from high-resource languages with millions of tokens to endangered languages with only a few thousand annotated words.

Importantly, UD includes a Yoruba treebank (Yoruba-YTB), compiled by Ishola et al. (2020), containing 317 sentences and 8,243 tokens. This treebank provides gold-standard annotations perfectly suitable for training and evaluating POS taggers. Frustratingly however, the existence of this invaluable resource has not at all led to widespread support for Yoruba in popular NLP libraries. Stanza, one of the most widely used academic NLP toolkits, as of version 1.11.0, provides pretrained models for 95 languages, yet excludes Yoruba despite the availability of UD training data. This gap between resource creation and infrastructure support exemplifies the structural barriers facing low-resource languages even when linguistic expertise and annotation labor have been invested.

3.4 Gaps in Current Research

While there has been substantial work done to address low-resource language NLP and dialectal variation separately, few studies have directly compared performance across these dimensions using consistent evaluations. Most multilingual benchmarks focus on languages with existing pretrained models, excluding low-resource languages like Yoruba where infrastructure gaps prevent evaluation entirely. Similarly, dialectal variation research has largely focused on English varieties or major European languages, with Caribbean Spanish and other Global South dialects receiving limited attention.

This study addresses these gaps by conducting parallel evaluations across high-resource languages (English, Standard Spanish), a low-resource language with available gold-standard data but no pretrained models (Yoruba), and a dialectal variety lacking both infrastructure support and substantial corpora (Dominican Spanish). By using identical evaluation methods and the UD framework, I provide direct quantitative comparison of performance disparities while addressing the qualitative patterns of errors that reveal the underlying biases in NLP architecture.

4. Data

4.1 Universal Dependencies Test Sets

All evaluations in this experiment use test sets from the UD v2.17 release. UD corpora are split into training, development, and test sets, with the test set reserved for final evaluation to ensure models are assessed on previously unseen data. For this study, I use only the test sets, as I am evaluating pretrained models rather than training new ones. This approach simulates real-world deployment conditions where users apply existing tools to new text.

English-EWT (English Web Treebank): The English test set comprises 2,077 sentences totaling 25,094 tokens. This corpus includes text from blogs, newsgroups, emails, reviews, and question-answer forums, providing a representative sample of informal written English from web sources. The genre diversity is particularly important because it ensures the evaluation includes conversational registers rather than only formal edited text.

Spanish-GSD (Spanish Google Universal Dependencies): The Spanish test set contains 427 sentences with 12,002 tokens. This corpus derives from Spanish Wikipedia, news articles, and web texts, primarily representing Peninsular Spanish varieties with some Latin American influence. While the corpus is labeled as "Spanish" rather than specifying regional varieties, its sources suggest a bias toward formal, edited text in relatively standardized varieties.

Yoruba-YTB (Yoruba Treebank): The Yoruba test set includes 317 sentences totaling 8,243 tokens. This corpus draws from a variety of written Yoruba sources including news, creative writing, and religious texts. Yoruba orthography includes three tone diacritics (low, mid, and high tone), though these are not consistently marked in all texts. The treebank represents Standard Yoruba as used in Nigeria.

All three corpora use the UD v2 annotation scheme, ensuring consistency in POS tags, morphological features, and dependency relations. This standardization is essential for cross-linguistic comparison, as it eliminates inconsistencies from different annotation conventions.

4.2 Dominican Spanish Corpus Construction

No publicly available annotated corpus of Dominican Spanish exists. Thus, in order to evaluate model performance on this dialectal variety I constructed a custom evaluation corpus of 150 sentences containing 870 tokens. This corpus was generated programmatically using templates, and then followed by manual review by a native speaker, myself, to ensure linguistic naturalness and coverage of dialectal features. The generation process proceeded as follows:

Step 1: Feature Inventory

I identified four categories of distinctive Dominican Spanish features:

- Phonological contractions: *ta* (está = is), *pa'* (para = for), *to'* (todo = all, each, everything)
- Lexical items: *vaina* (thing/stuff), *tiguere* (smart ass, cunning guy), *colmado* (corner store), *guagua* (bus)
- Code-switches: English borrowings common in Dominican speech, particularly in urban and diaspora contexts (*ticket*, *meeting*, *parking*, *full*)
- Regional vocabulary: *sancocho* (stew), *motoconcho* (motorcycle taxi), *chin* (a little bit)

Step 2: Template Design

I created sentence templates representing common syntactic patterns in Dominican Spanish. Each template specifies slots for different word categories (QU_WORD, PRON, VERB, NOUN, ADJ, etc.) along with their correct POS tags.

```
Template: [("¿", "PUNCT"), ("QU_WORD", "PRON/ADV"), ("PRON", "PRON"),
           ("ta", "VERB"), ("ADJ", "ADJ"), ("?", "PUNCT")]
Generated: "¿Cómo tú ta?" (How are you?)
```

Step 3: Lexicon Population.

For each slot type, I created a word bank containing appropriate lexical items with their correct POS tags. This approach allows systematic variation while maintaining grammatical coherence.

```
"QU_WORD": [
    ("qué", "PRON"), ("cómo", "ADV"), ("dónde", "ADV"),
    ("cuándo", "ADV"), ("por qué", "ADV")
]
```

Step 4: Sentence Generation.

The generator randomly selects templates and fills slots from the appropriate word banks, creating varied sentences that include dialectal features. For example:

```
"Necesito un ticket pa' el trabajo."
(I need a ticket for work.)
POS: [VERB DET NOUN ADP DET NOUN PUNCT]

"Ese tiguere ta enratiao."
(That guy is angry.)
POS: [DET NOUN AUX ADJ PUNCT]
```

Step 5: Manual Review

As a heritage speaker of Dominican Spanish, I reviewed all programmatically generated sentences for naturalness and grammatical acceptability. The resulting 150-sentence corpus includes 71 instances of *pa'* (the most frequent contraction), 67 instances of *ta*, multiple code-switches, and representative samples of unique lexical items. While this corpus is substantially smaller than the other test sets (870 tokens vs. 8,000+), it provides enough data to identify systematic error patterns and measure overall accuracy in how standard models fare for

dialects. The programmatic generation approach, while definitely unconventional, addresses a fundamental barrier in dialectal variation research: the absence of annotated data for non-standard varieties.

4.3 Data Limitations and Considerations

Several limitations of the data should be noted. First, the corpus sizes vary substantially, from 870 tokens (Dominican Spanish) to 25,094 tokens (English). This makes direct comparison of raw error counts unsuitable, though accuracy percentages and per-POS statistics remain comparable. The controlled experiment (which will be expanded upon in Section 5.2) tries to address this concern by limiting all corpora to 5,000 tokens, however that data gap still persists.

The Dominican Spanish corpus, while informed by my native speaker intuition, was still generated programmatically rather than collected from naturalistic speech or writing. This introduces potential bias toward simpler syntactic structures and may not fully capture the discourse-level features of true Dominican Spanish. However, for the specific purpose of evaluating POS tagging on unique dialectal features, the corpus provides a controlled test set that would be difficult to extract from naturally occurring text considering the complete absence of pre-existing annotations.

All corpora represent written language rather than spoken transcription. Phonological processes that appear in writing (like *pa'* for *para*) are captured, but features that exist primarily in speech but not orthography are not included. This limitation affects all varieties, though it may be particularly significant for dialectal varieties where written standards often diverge from actual usage.

The Standard Spanish corpus, while labeled generically as “Spanish,” reflects primarily Peninsular and formal Latin American varieties. Caribbean Spanish features may appear only sporadically or not at all in this corpus, making the comparison with Dominican Spanish a comparison between relatively formal/standard written Spanish and informal/colloquial dialectal Spanish. This inconsistency of register and dialect is difficult to avoid given available resources.

Corpus	Sentences	Tokens
English EWT	2077	25094
Spanish GSD	427	12002
Yoruba YTB	317	8243
Programmatically Generated Dominican Spanish	150	870

5. Methodology

5.1 Tool Selection and Rationale

This study uses Stanza (Qi et al., 2020) as the primary POS tagging tool. Stanza is a Python library developed by the Stanford NLP Group that provides neural network-based NLP analysis for 95 languages at the time of writing. Importantly, Stanza models are trained directly on Universal Dependencies corpora, ensuring consistency between training data and evaluation framework. This design makes Stanza particularly suitable for cross-linguistic research, as it eliminates any potential inconsistencies from different annotation conventions or training data sources.

Stanza's architecture uses a bidirectional LSTM (Long Short-Term Memory) neural network with character-level representations to capture morphological patterns and a word-level embedding layer pretrained on language-specific corpora. For POS tagging specifically, the model processes tokenized input and assigns each token a UD tag based on learned patterns from its training corpus. The model's neural design means it learns representations implicitly rather than using hand-crafted features, making it a suitable representative of current NLP approaches.

As for alternative tools, I initially planned to evaluate both Stanza as well as spaCy, another popular NLP library. However, preliminary testing revealed a fundamental incompatibility: spaCy uses proprietary tokenization schemes that differ from Universal Dependencies. For the English test set, spaCy's tokenizer produced 25,728 tokens compared to UD's 25,094 tokens, a discrepancy of 634 tokens (2.5%). This mismatch makes direct comparison impossible, as predicted tags cannot be aligned one-to-one with gold-standard tags. For instance, spaCy might split hyphenated words like “you-know-who” into five tokens while UD treats it as three.

This tokenization issue highlights a broader methodological challenge in NLP evaluation, as pretrained models often embed preprocessing decisions that are not at all transparent to end-users and incompatible with standard benchmarks. While possible workarounds exist, such as post-hoc alignment algorithms using edit distance, not only are they much beyond my current expertise as an undergraduate student, but also they introduce scope creep, potential error sources, and additional complexity to a project that is already complex as-is. For this project, I prioritized evaluation validity over tool diversity, using only Stanza for the final experiment.

5.2 The Yoruba Infrastructure Gap

An important finding emerged during the experimental setup: Stanza does not provide pretrained models for Yoruba despite the language's 48 million L1 speakers and the existence of a Yoruba treebank in UD. When attempting to load a Yoruba model, Stanza returns an error:

```
import stanza
nlp = stanza.Pipeline('yo')
ValueError: No processors to load for language yo. Language yo is
currently unsupported

Supported languages: af, ang, ar, be, bg, bn, bxr, ca, cop, cs, cu,
cy, da, de, el, en, es, et, eu, fa, fi, fo, fr, fro, ga, gd, gl, got,
grc, gv, hbo, he, hi, hr, hsb, hu, hy, hyw, id, is, it, ja, ka, kk,
kmr, ko, kpv, ky, la, lij, lt, lv, lzh, ml, mr, mt, multilingual, my,
myv, nb, nds, nl, nn, or, orv, ota, pcm, pl, pt, qaf, qpm, qtd, ro,
ru, sa, sd, si, sk, sl, sme, sq, sr, sv, ta, te, th, tr, ug, uk, ur,
vi, wo, xcl, zh-hans, zh-hant
```

This absence is particularly striking given that Stanza supports not just languages with far fewer speakers but also historical languages, including but not limited to: Gothic (got), Old Church Slavonic (cu), Old French (fro), and Ancient Greek (grc). The inclusion of historical languages with zero native speakers while excluding a low-resource language with 48 million L1 speakers reveals that model availability is not determined at all by linguistic importance or speaker population. Rather, it is dictated by the research priorities and resource allocation of NLP labs, which are disproportionately located in wealthy Western institutions and serve European languages.

To quantify the impact of this infrastructure gap, I implemented a naive baseline tagger that predicts NOUN for every token. This baseline represents the performance floor achievable by simply guessing the most frequent POS tag without any linguistic knowledge:

```
# Naive baseline: predict NOUN for all tokens
# NOUN is the most frequent POS tag in the Yoruba treebank

def baseline_yoruba_tagger(tokens):
    predictions = []
    for token in tokens:
        predictions.append("NOUN")
    return predictions
```

This baseline serves three purposes: (1) it demonstrates what any user would achieve with zero support in their native language, (2) it provides a reference point for evaluating future Yoruba models, and (3) it undeniably quantifies the cost of systematic exclusion from NLP tools. The resulting accuracy of 18.3% starkly illustrates the consequences of infrastructure gaps.

5.3 Evaluation Procedure

The evaluation stage proceeds in three parts: preprocessing, tagging, and metric calculation.

Stage 1: Preprocessing and Data Loading

UD corpora are distributed in CoNLL-U format, a tab-separated plain text format where each line represents a token with ten fields including word form, POS tag, and dependency information. I wrote a parser to extract tokens and their gold-standard POS tags.

```
# Parse a CoNLL-U file, extract tokens with gold POS tags, and
return:
# Sentences: List of sentence objects, each containing tokens and
```

```
tags
# total_tokens: Total number of tokens in the corpus

def load_conllu(filepath):

    sentences = []
    current_sentence = {"tokens": [], "tags": []}

    with open(filepath, 'r', encoding='utf-8') as f:
        for line in f:
            line = line.strip()

            # Skips comments
            if line.startswith('#'):
                continue

            # An empty line indicates sentence boundary
            if not line:
                if current_sentence["tokens"]:
                    sentences.append(current_sentence)
                    current_sentence = {"tokens": [], "tags": []}
                continue

            # Parses token line
            fields = line.split('\t')
            token_id = fields[0]

            # Skips multi-word tokens
            if '-' in token_id or '.' in token_id:
                continue

            token = fields[1]  # Word form
            upos = fields[3]   # Universal POS tag

            current_sentence["tokens"].append(token)
            current_sentence["tags"].append(upos)

    # Adds final sentence if file doesn't end with blank line
    if current_sentence["tokens"]:
```

```
        sentences.append(current_sentence)

    total_tokens = sum(len(sent["tokens"]) for sent in sentences)

    return sentences, total_tokens
```

This parser handles several CoNLL-U format details: skipping comment lines, detecting sentence boundaries, and excluding multi-word tokens (which have IDs like '1-2' representing contractions split across multiple syntactic words). The result is a list of sentence objects, each containing parallel lists of tokens and their gold-standard tags.

Stage 2: POS Tagging

For each language, I initialize a Stanza pipeline and process all sentences:

```
import stanza

# Tags a corpus using Stanza's pretrained models
# Returns a list of predicted POS tags parallel to gold tags

def tag_corpus_stanza(sentences, language_code):

    # Initialize Stanza pipeline
    # tokenize_pretokenized=True uses our existing tokenization

    nlp = stanza.Pipeline(
        language_code,
        processors='tokenize,pos',
        tokenize_pretokenized=True
    )

    predictions = []

    for sentence in sentences:
```

```
        # Stanza expects pre-tokenized input as list of tokens
        doc = nlp([sentence["tokens"]])

        # Extract predicted tags
        for sent in doc.sentences:
            for word in sent.words:
                predictions.append(word.upos)

    return predictions
```

The “tokenize_pretokenized=True” parameter is absolutely important here as it instructs Stanza to use our existing tokenization from the UD corpus rather than automatically applying its own tokenizer. This ensures perfect alignment between predictions and gold tags, eliminating tokenization mismatches as a potential source of error.

Meanwhile, for Yoruba, this tagging stage is replaced with the baseline tagger, since no Stanza model exists. As for Dominican Spanish, the Standard Spanish model is utilized here to test how it handles dialectal variety. For English and Spanish, the pipeline downloads pretrained models automatically on first use, and models are then cached locally for subsequent runs.

Stage 3: Metric Calculation

I compute three primary metrics: overall accuracy, macro-averaged F1 score, and weighted-averaged F1 score.

Accuracy is the simplest metric, measuring the percentage of tokens assigned the correct POS tag:

```
from sklearn.metrics import accuracy_score

# Calculates percentage of correctly tagged tokens
# Formula: accuracy = (correct predictions) / (total predictions)

def calculate_accuracy(gold_tags, predicted_tags):
    return accuracy_score(gold_tags, predicted_tags)
```

F1 Score balances precision (what percentage of predicted tags are correct) and recall (what percentage of gold tags are predicted). For multi-class classification like POS tagging, we can calculate F1 in two ways:

```
from sklearn.metrics import f1_score

# Calculate F1 scores using two averaging methods
# Macro F1: Calculates F1 for each POS tag separately, then average.
Treats all tags equally regardless of frequency
# Weighted F1: Calculates F1 for each tag, then average weighted by
tag frequency. Emphasizes performance on common tags.


def calculate_f1_scores(gold_tags, predicted_tags):
    f1_macro = f1_score(
        gold_tags,
        predicted_tags,
        average='macro',
        zero_division=0
    )

    f1_weighted = f1_score(
        gold_tags,
        predicted_tags,
        average='weighted',
        zero_division=0
    )
```

```
    return f1_macro, f1_weighted
```

Macro F1 treats all POS tags equally, making it sensitive to performance on rare categories like interjections or particles. Weighted F1 gives more importance to frequent tags like nouns and verbs, better reflecting overall performance on typical text. Both metrics are equally as informative and important, as any large gaps between them can indicate that a model performs well on common tags but struggles on rare ones, or vice versa.

5.4 Error Analysis

Beyond accuracy metrics, I conducted detailed error analysis using confusion matrices and per-POS accuracy breakdowns. A confusion matrix shows which POS tags are mistaken for which others.

```
from sklearn.metrics import confusion_matrix
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Generates and visualizes a confusion matrix showing error patterns
# Rows: gold (true) tags
# Columns: predicted tags
# Cell (i,j): number of times gold tag i was predicted as tag j

def create_confusion_matrix(gold_tags, predicted_tags,
language_name):

    # Gets all unique tags that appear in either gold or predictions
    all_tags = sorted(list(set(gold_tags + predicted_tags)))

    # Computes the confusion matrix
    cm = confusion_matrix(
        gold_tags,
```

```
        predicted_tags,
        labels=all_tags
    )

    # Creates the heatmap
    plt.figure(figsize=(12, 10))
    sns.heatmap(
        cm,
        annot=True, # Shows numbers in cells
        fmt='d', # Formats as integers
        cmap='Blues', # Color scheme
        xticklabels=all_tags,
        yticklabels=all_tags
    )

    # Labelling
    plt.xlabel('Predicted Label')
    plt.ylabel('Gold Label')
    plt.title(f'Confusion Matrix - {language_name}')

    plt.tight_layout()
    plt.savefig(f'confusion_matrix_{language_name.lower()}.png',
dpi=300) # Saves as png
    plt.close()

    return cm, all_tags
```

Alongside that, I computed per-POS accuracy to identify which grammatical categories are most challenging.

```
# Calculates accuracy separately for each POS tag
# Returns dictionary of {tag: accuracy} for all tags that appear in
gold data

def calculate_per_pos_accuracy(gold_tags, predicted_tags):

    per_pos_accuracy = {}
```

```
    # Gets unique tags
    unique_tags = set(gold_tags)

    for tag in unique_tags:

        # Finds all positions where this tag appears in gold data
        tag_indices = [i for i, t in enumerate(gold_tags) if t ==
tag]

        # Counts correct predictions for this tag
        tag_correct = sum(
            1 for i in tag_indices
            if predicted_tags[i] == tag
        )

        # Calculates accuracy for this tag
        tag_total = len(tag_indices)
        per_pos_accuracy[tag] = tag_correct / tag_total if tag_total
> 0 else 0

    return per_pos_accuracy
```

I also tracked the most frequently misclassified tokens to identify specific lexical items causing errors:

```
from collections import Counter

# Identifies which specific words are misclassified most often
# Returns a counter object mapping (token, gold_tag, predicted_tag)
to count

def find_most_misclassified_tokens(sentences, gold_tags,
predicted_tags):

    errors = []
    token_index = 0
```

```
    for sentence in sentences:
        for token in sentence["tokens"]:
            if gold_tags[token_index] != predicted_tags[token_index]:
                error_triple = (
                    token,
                    gold_tags[token_index],
                    predicted_tags[token_index]
                )
                errors.append(error_triple)
            token_index += 1

    error_counts = Counter(errors) # Counts frequency of each error
type

    return error_counts.most_common(10)  # Returns top 10 errors
```

For Dominican Spanish, this analysis is important as it reveals whether errors are concentrated on specific dialectal features (like *pa'* or *ta*) or distributed across the lexicon.

5.5 Controlled Experiment Design

The initial pretrained evaluation compares models as used in practice, with corpus sizes reflecting current data availability. However, this introduces a potentially inconsistent variable: differences in accuracy might reflect differences in evaluation set size rather than true performance disparities. Larger corpora provide more stable accuracy estimates and include more diverse linguistic constructions, potentially making high-resource languages appear to perform better simply because we have more data to evaluate them on.

To address this, I conducted a second experiment equalizing corpus sizes. I limited each language to exactly 5,000 tokens (except Dominican Spanish, which only has 870 tokens available):

```
# Samples sentences from corpus until reaching target token count
# Takes sentences in order (not randomly) to preserve the corpus' structure.

def sample_corpus(sentences, target_tokens=5000):

    sampled_sentences = []
    token_count = 0

    for sentence in sentences:
        sentence_length = len(sentence["tokens"])

        # Stops if adding this sentence would exceed target
        if token_count + sentence_length > target_tokens:
            break

        sampled_sentences.append(sentence)
        token_count += sentence_length

    return sampled_sentences, token_count
```

This sampling is deterministic rather than random and takes sentences in their original corpus order. This approach preserves the genre distribution and discourse structure of the corpus rather than creating a random sample that might over-represent short sentences or certain syntactic patterns.

The controlled experiment answers a specific question: if we evaluate all languages on equal amounts of data, do performance disparities persist? If accuracy gaps narrow substantially in the

controlled condition, it would suggest that the initial disparities reflected evaluation set size rather than fundamental model limitations. If gaps persist or widen, it indicates that the problems are architectural and data-independent.

5.6 Implementation Details

All code was implemented in Python 3.10.11 on a Windows 11 operating system using the following library versions:

- stanza 1.11.0
- scikit-learn 1.7.2
- pandas 2.3.3
- matplotlib 3.10.7
- seaborn 0.13.2
- numpy 2.2.6

The full experimental pipeline consists of ~800 lines of Python code organized into the following modules:

pos_evaluation.py (~600 lines): The main experimental script that orchestrates the entire evaluation pipeline. This script handles: (1) CoNLL-U file parsing and corpus loading, (2) Stanza pipeline initialization and POS tagging (with baseline implementation for Yoruba), (3) accuracy and F1 score calculation using scikit-learn, (4) confusion matrix generation and visualization, (5) per-POS accuracy breakdown, (6) identification of most frequently misclassified tokens, and (7) controlled experiment corpus sampling. The script is structured as a

series of functions that process each language sequentially, generating CSV files with results and PNG files with visualizations.

dominican_corpus.py (~150 lines): A class-based generator that creates the Dominican Spanish evaluation corpus using templates. The class defines dialectal features, sentence templates, and a word bank, then generates sentences by randomly selecting templates and filling slots with appropriate lexical items. The generator uses random.choice() for template and word selection, producing 150 sentences with 870 tokens that include phonological contractions (*pa'*, *ta*), code-switches (*meeting*, *parking*, *full*), and distinctive Dominican vocabulary (*tiguere*, *vaina*, *enratiao*). Generated sentences are exported to CoNLL-U format for evaluation.

Experimental runtime varies by language due to differences in corpus size and model complexity. The majority of time is spent waiting for Stanza rather than in data loading or metric computation.

6. Results

6.1 Pretrained Model Performance

The real-world evaluation reveals considerable performance disparities across languages:

Language	Accuracy	F1 Macro	F1 Weighted	Total Tokens
English-Stanza	0.9654	0.9169	0.9649	25094
Spanish-Stanza	0.9293	0.7707	0.9281	12002

Yoruba-Stanza	0.1833	0.0182	0.0568	8243
Dominican Spanish-Stanza	0.7345	0.4533	0.7522	870

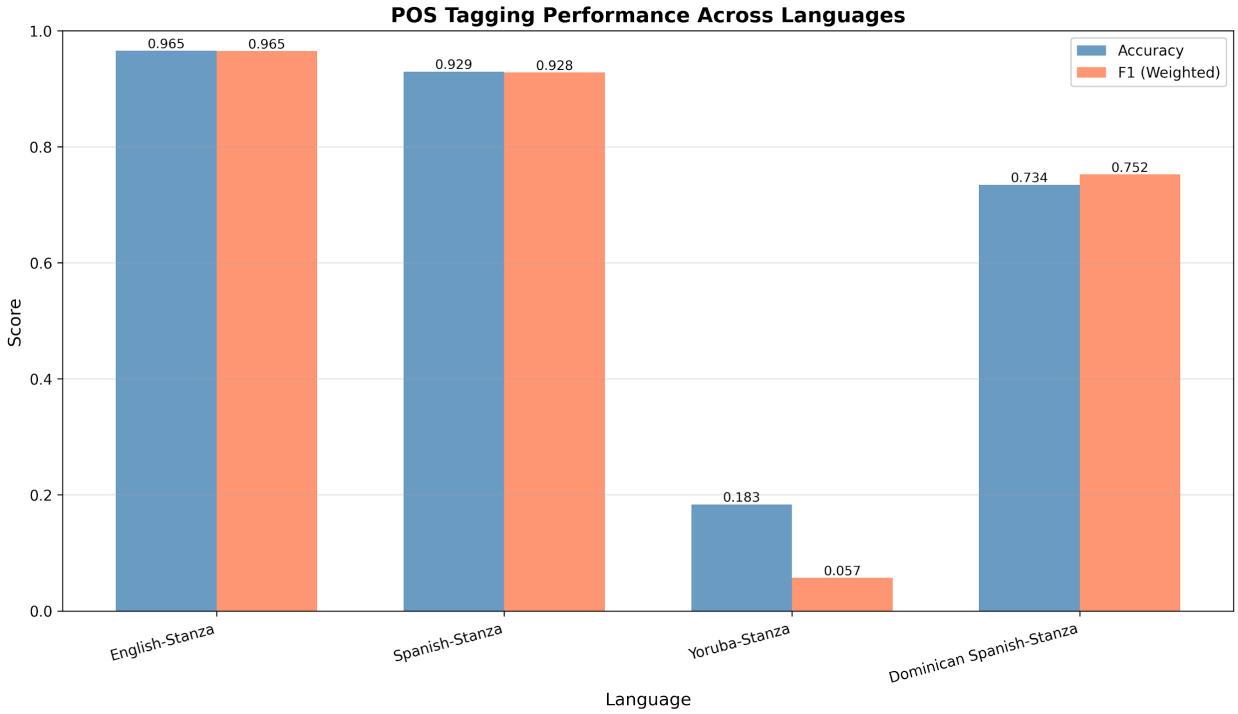
English achieves the highest accuracy at 96.54%, with a weighted F1 of 0.9649 closely matching this value. The macro F1 of 0.9169 is somewhat lower, showing reduced performance on rare POS categories, but remains strong overall. With 868 errors out of 25,094 tokens (3.46% error rate), the model performs at near-human levels for this high-resource language.

Standard Spanish performs nearly as well, achieving 92.93% accuracy with a weighted F1 of 0.9281. The macro F1 of 0.7707 is notably lower than English's, suggesting greater difficulty with infrequent tags. The model makes 848 errors across 12,002 tokens (7.07% error rate), approximately twice English's error rate but still within acceptable ranges for most applications.

Yoruba expectedly achieves only 18.33% accuracy, with extremely low F1 scores of 0.0182 (macro) and 0.0568 (weighted). This represents complete model failure with 6,732 errors out of 8,243 tokens (81.67% error rate). Recall that this “model” is actually the naive baseline that predicts NOUN for every token, so these numbers reflect the consequence of NLP exclusion rather than actual NLP performance. The weighted F1 being higher than macro F1 indicates that the baseline succeeds more often on frequent tags, as NOUN is common.

Dominican Spanish achieves 73.45% accuracy with weighted F1 of 0.7522 and macro F1 of 0.4533. This represents 231 errors across 870 tokens (26.55% error rate), approximately 3.7 times higher than Standard Spanish's error rate. The large gap between weighted and macro F1

(0.7522 vs. 0.4533) suggests severe problems with specific POS categories, particularly ones that are frequent in Dominican Spanish but rare or absent in Standard Spanish training data.

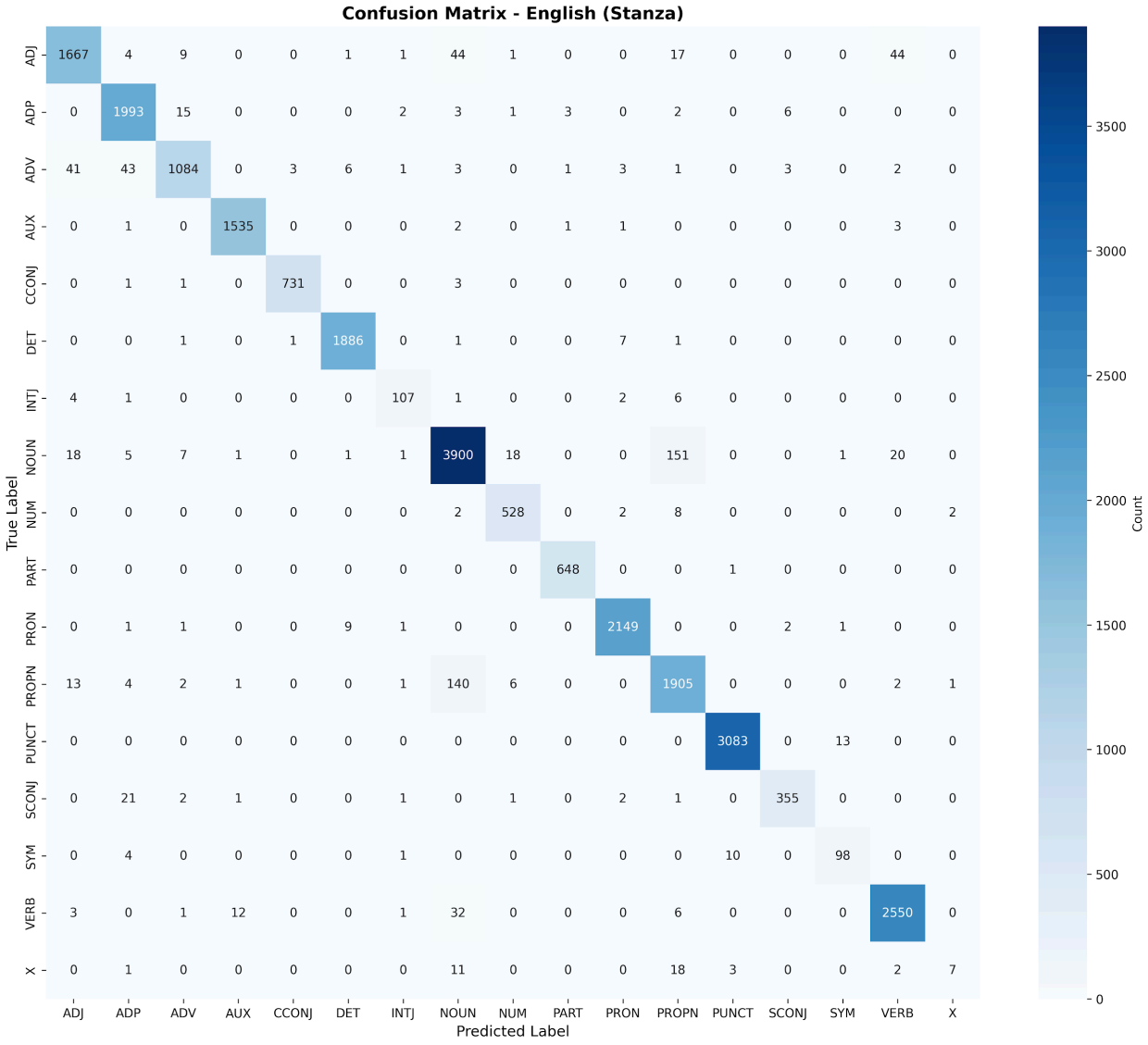


The accuracy gap between highest-performing (English: 96.54%) and lowest-performing (Yoruba: 18.33%) languages is 78.21 percentage points. Even excluding Yoruba (which lacks any real model), the gap between English and Dominican Spanish is 23.09 percentage points, while the gap between Standard Spanish and Dominican Spanish is 19.48 percentage points. These disparities far exceed the margin of error from corpus size differences and demonstrate systematic infrastructure bias.

6.2 Error Patterns: High-Resource Languages

The confusion matrices for English and Spanish reveal the types of errors that occur even for well-supported languages.

English Error Analysis



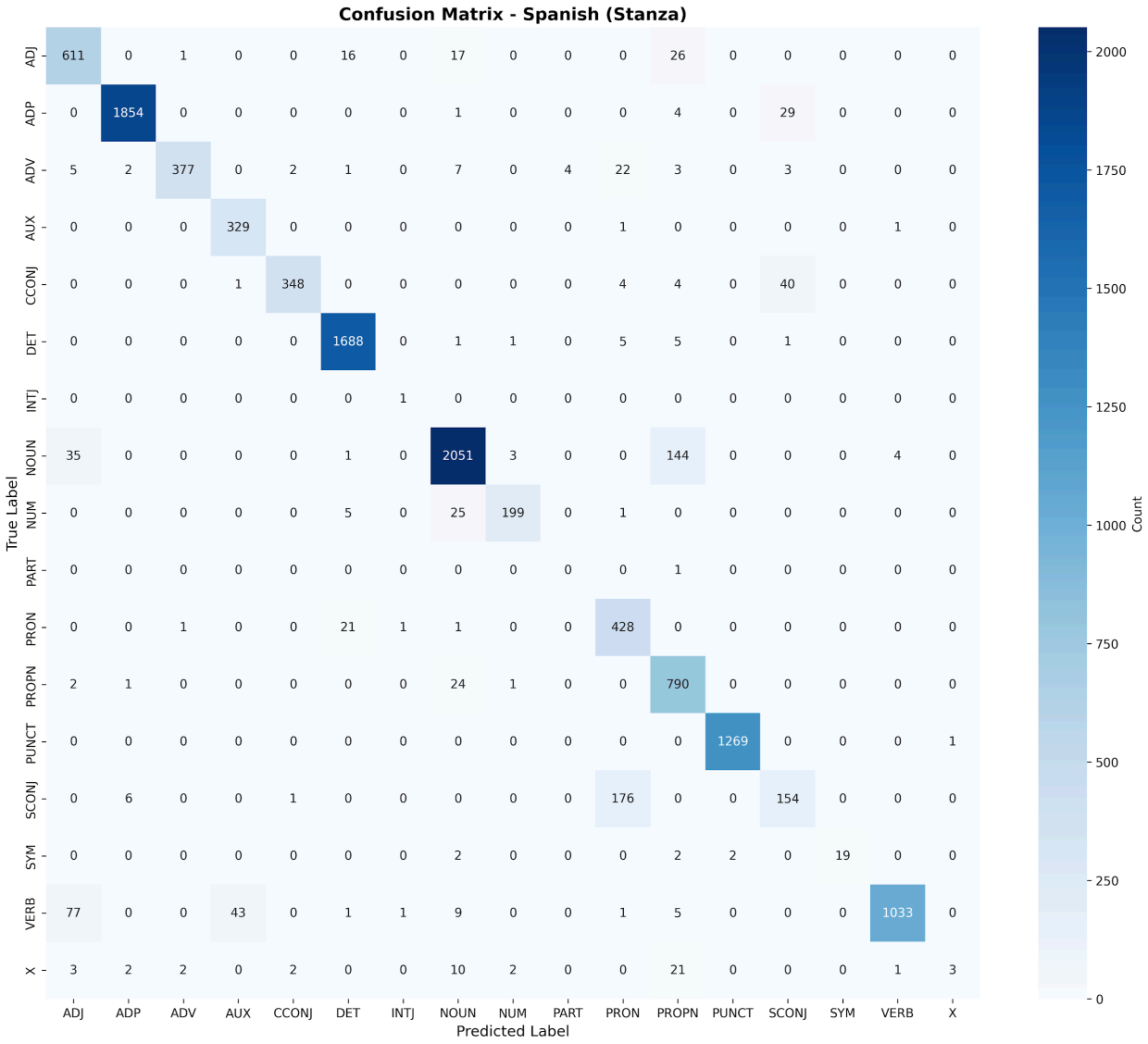
The most frequent English confusions are:

1. NOUN ↔ PROPN: 151 errors of NOUN predicted as PROPN, 140 errors in reverse direction. This confusion is unsurprising because the distinction often depends on world knowledge (is Apple a fruit or a company?) or discourse context (first mention vs. subsequent mention of entities).
2. ADJ > VERB: 44 errors. English adjectives and past participles are often homophonous (e.g., “broken” in “the broken window” vs. “has broken”), requiring syntactic context to disambiguate.
3. ADV > ADP: 43 errors. Many English words function as both adverbs and prepositions depending on whether they take objects (e.g., “up” in “look up” (ADV) vs. “up the stairs” (ADP)).

The most frequently misclassified tokens include well known ambiguities in English:

- out: 14 errors (ADV/ADP ambiguity)
- in: 14 errors (ADV/ADP ambiguity)
- up: 12 errors (ADV/ADP ambiguity)
- to: 10 errors (ADP/PART ambiguity in infinitives)

Spanish Error Analysis



Spanish errors show similar patterns but with some language-specific characteristics:

1. SCONJ > PRON: 176 errors. This confusion almost entirely involves *que*, which functions as both a subordinating conjunction (*"Creo que…"* = "I think that…") and a relative pronoun (*"El libro que leí."* = "The book that I read."). Spanish *que* is notoriously difficult even for native speakers to categorize in some contexts.
2. NOUN ↔ PROPN: 144 errors, similar situation to English.

3. VERB → ADJ: 77 errors, reflecting participle ambiguity similar to English.

The token “que” accounts for 209 of the most frequently misclassified items, dominating the error distribution. This single highly ambiguous word creates the majority of Spanish errors, suggesting that overall performance is quite good aside from this systematic challenge.

English and Spanish: Summary

For both English and Spanish, per-POS accuracy analysis shows high performance across most categories:

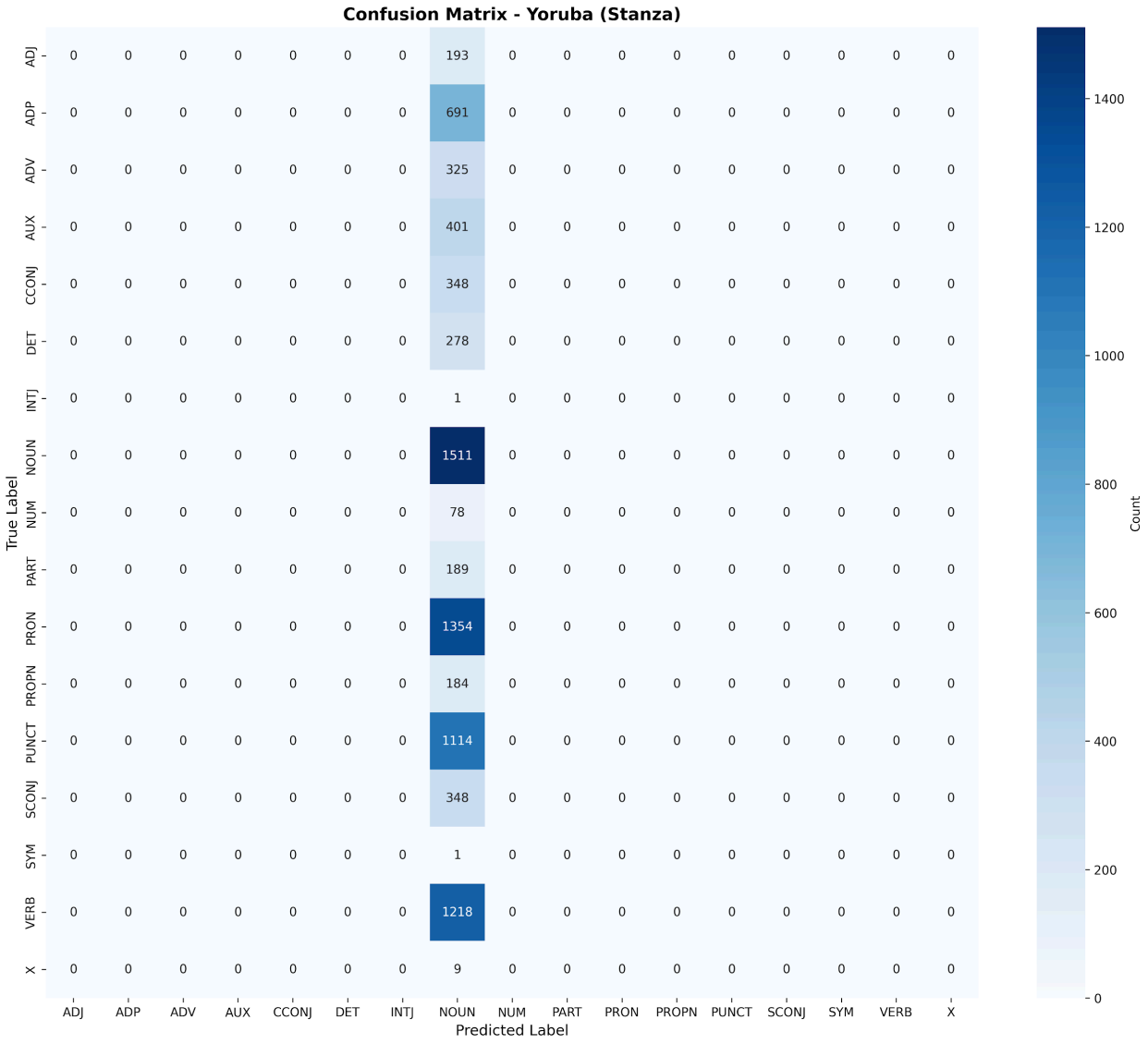
- PUNCT: ~100% accuracy in both languages
- DET: 98-99% accuracy
- PRON: 95-99% accuracy
- NOUN, VERB, ADJ: 85-95% accuracy

Lower performance appears on:

- SCONJ: 46% in Spanish (due to *que*), 92% in English
- X: 17% in English, 7% in Spanish

These patterns indicate that models trained on high-resource languages achieve strong performance across most grammatical categories, with errors concentrated on inherently ambiguous constructions rather than systematic mishandling of dialectal or non-standard features.

6.3 Yoruba: Infrastructure Failure



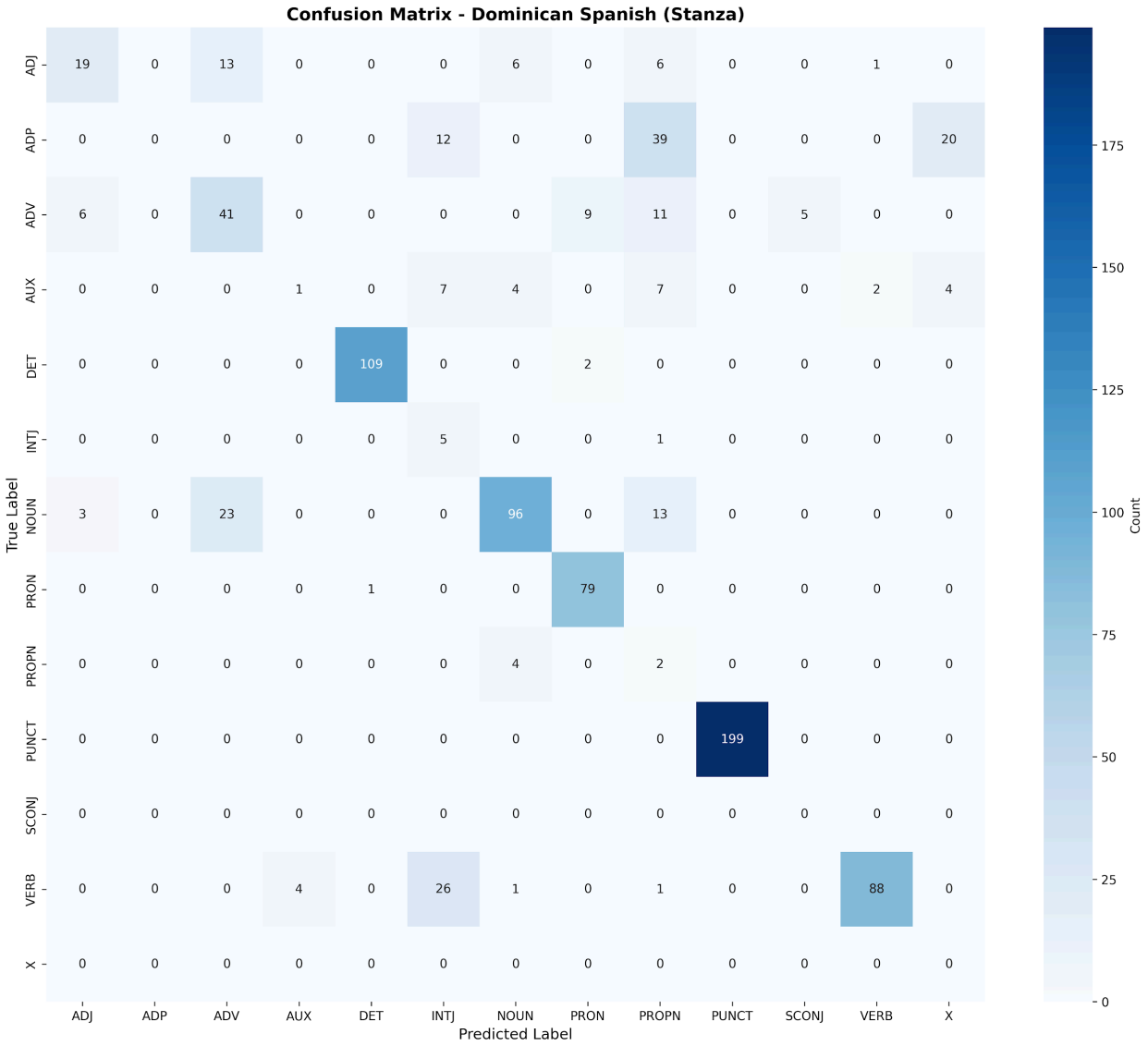
The Yoruba confusion matrix reveals complete infrastructure failure. Because the baseline tagger predicts NOUN for every token, the matrix shows a single vertical stripe in the NOUN column. Every gold tag, regardless of what it actually is, receives the prediction NOUN.

The most frequent errors are:

1. PRON > NOUN: 1,354 errors (all pronouns tagged as nouns)
2. VERB > NOUN: 1,218 errors (all verbs tagged as nouns)
3. PUNCT > NOUN: 1,114 errors (all punctuation tagged as nouns)

The baseline achieves 18.33% accuracy solely because NOUN is the most frequent tag in Yoruba, representing pure chance alignment rather than any linguistic analysis. This result quantifies the cost of systematic exclusion. Yoruba speakers attempting to use POS tagging tools would encounter complete failure, not slightly degraded performance. The gap between Yoruba's baseline (18.33%) and English's pretrained performance (96.54%) represents a 78.2 percentage point difference attributable entirely to infrastructure investment rather than linguistic properties of the languages themselves.

6.4 Dominican Spanish: Systematic Dialectal Bias



Dominican Spanish reveals a different failure, as the model works for standard features but systematically breaks on dialectal ones. The confusion matrix shows concentrated errors on specific categories:

1. ADP > PROPN: 39 errors (prepositions tagged as proper nouns)
2. VERB > INTJ: 26 errors (verbs tagged as interjections)

3. NOUN > ADV: 23 errors (nouns tagged as adverbs)
4. ADP > X: 20 errors (prepositions tagged as unknown)
5. ADP > INTJ: 12 errors (prepositions tagged as interjections)

Note that ADP appears in three of the top five error types, accounting for 71 errors total. This pattern directly reflects the contracted preposition *pa'* which the model consistently fails to recognize.

Token	**POS**	**# of Times Misclassified**
pa’	ADP	71
ta	VERB/AUX	56
enratiao	ADJ	13
bien	ADV	13
mañana	NOUN/ADV	11
full	ADJ/ADV	10
hoy	NOUN/ADV	10
cómo	ADV	9
rápido	ADJ/ADV	7
por qué	ADV	5

Some interesting findings to note regarding this table:

- pa': 71 errors (gold: ADP, predicted: PROPN/X/INTJ). The model does not recognize this contracted form of *para* and variously tags it as a proper noun, unknown word, or interjection.

- ta: 56 errors (gold: VERB/AUX, predicted: INTJ/NOUN). The model does not recognize this contracted form of *está* and treats it as an interjection or noun rather than a verb/auxiliary.
- enratiao: 13 errors (gold: ADJ, predicted: varies). This distinctively Dominican adjective (meaning angry/annoyed) does not appear in Standard Spanish training data and is systematically misclassified.
- full: 10 errors (gold: ADJ/ADV, predicted: varies). This English code-switch is common in Dominican Spanish (e.g., "Estaba full." = "It was packed.") but the Standard Spanish model cannot determine its grammatical category.

These four lexical items, all dialectal features, account for 150 of the 231 total errors (64.9%). The errors are not randomly distributed across the vocabulary but concentrated on precisely the features that distinguish Dominican Spanish from the standard varieties.

Dominican Spanish: Summary

Per-category accuracy reveals stark contrasts, compare high accuracy (standard features):

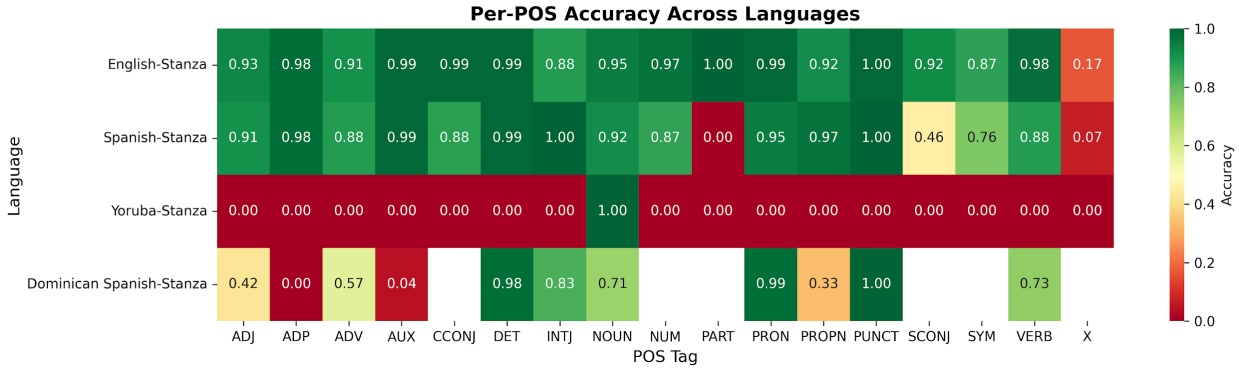
- PUNCT: 100% (punctuation works perfectly)
- DET: 98% (determiners recognized)
- PRON: 99% (pronouns recognized)
- NOUN: 71% (common nouns mostly work)

To near-zero accuracy (dialectal features):

- ADP: 0% (all contracted prepositions failed)

- AUX: 4% (contracted auxiliaries failed)
- PROPN: 33% (proper nouns confused with unknown words)

This selective failure pattern demonstrates that the model has learned Standard Spanish as the default and treats Dominican features as errors rather than valid linguistic variation. Standard grammatical categories (determiners, pronouns, basic nouns) work fine. Non-standard phonological, lexical, and code-switching features fail systematically.



6.5 Controlled Experiment Results

The controlled experiment equalized corpus sizes at 5,000 tokens per language (except Dominican Spanish, which remains at 870 tokens).

Language	Accuracy	F1 Macro	F1 Weighted	Total Tokens
English-Controlled	0.9740	0.8982	0.9740	5000
Spanish-Controlled	0.9322	0.7909	0.9312	5000
Yoruba-Controlled	0.1388	0.0163	0.0338	5000

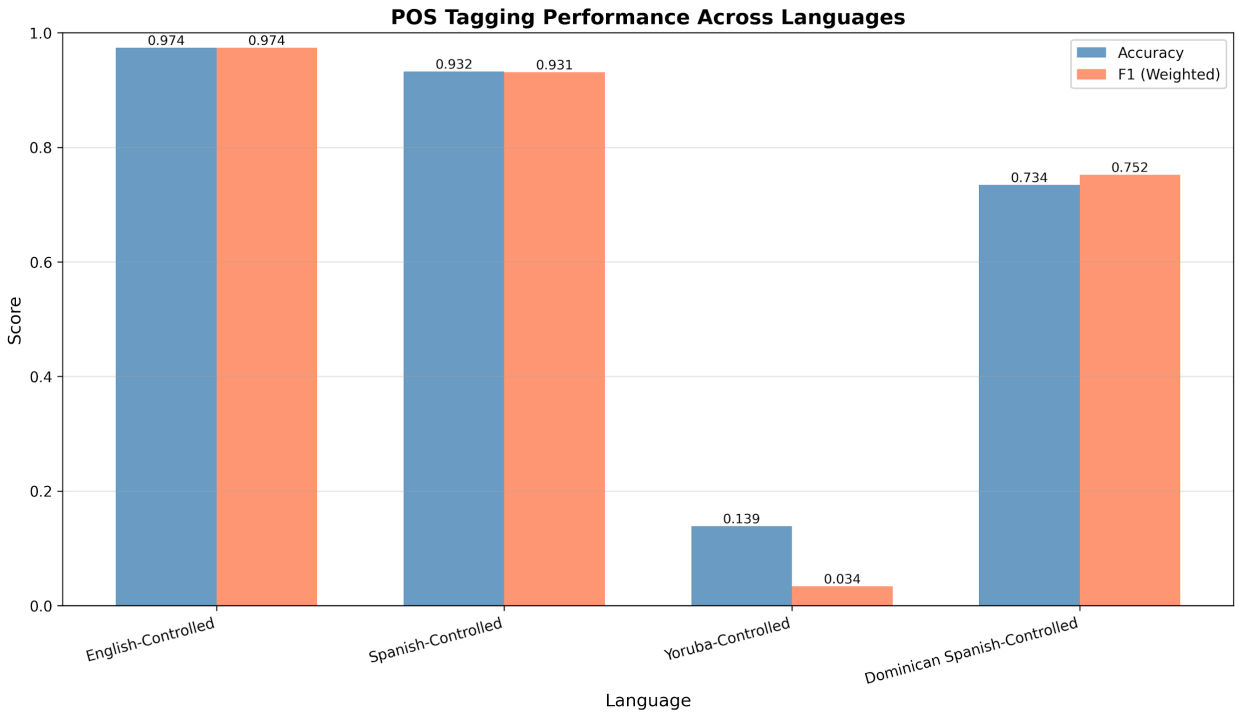
Dominican Spanish-Controlled	0.7345	0.4533	0.7522	870

English improved slightly from 96.54% to 97.40% (+0.86 percentage points). This small improvement may possibly reflect sampling variance: the first 5,000 tokens happened to include slightly fewer ambiguous constructions than the full corpus.

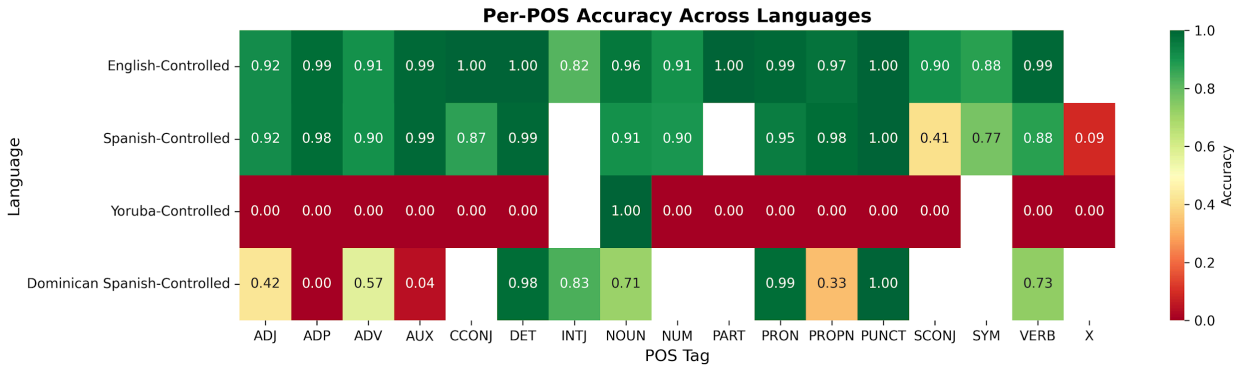
Spanish improved minimally from 92.93% to 93.22% (+0.29 percentage points), essentially remaining stable.

Yoruba decreased from 18.33% to 13.88% (−4.45 percentage points). The baseline performs worse on the limited sample, likely because the first 5,000 tokens contain proportionally fewer nouns than the full corpus.

Dominican Spanish remained identical at 73.45% because the full corpus contains only 870 tokens, less than the 5,000 token limit.



Interestingly, high-resource languages show slight improvements when evaluated on smaller corpora. This pattern may suggest that high-resource models benefit from transfer learning and robust representations that generalize well to small samples, while low-resource languages (lacking models) and dialectal varieties (trained on ill-suited data) cannot leverage similar advantages. The problem lies not in the quantity of evaluation data but in the availability of training data, pretrained models, and infrastructure support that benefits high-resource languages.



7. Discussion

7.1 Interpretation of Findings

This project provides quantitative evidence for three interrelated claims about language equity in NLP:

Claim 1: Low-resource languages face systematic infrastructure exclusion.

The 78 percentage point gap between English and Yoruba (in pretrained evaluation) represents not linguistic complexity or typological distance but complete absence of model support. Yoruba is a tonal language with a robust system of verbal extensions and serial verb constructions, but these features do not inherently make POS tagging more difficult than English's complex morphology and word order alternations. Rather, the absence of Yoruba models reflects priorities in NLP research that systematically deprioritize African languages despite their substantial speaker populations.

The existence of the Yoruba UD treebank makes this exclusion particularly unmistakable. The linguistic expertise and annotation labor required to create gold-standard training data have been

invested. The data exists in the standard format used by NLP tools. Yet, major libraries like Stanza do not provide pretrained models, creating a noticeable gap between resource creation and infrastructure deployment. This pattern suggests that language inclusion requires not just data collection but sustained advocacy, funding, and institutional commitment that African languages often lack compared to European ones.

Claim 2: Dialectal varieties encounter systematic bias even within their "own" language. The 19.48 percentage point accuracy gap between Standard and Dominican Spanish demonstrates that models encode standard language ideology. The error analysis reveals that this is not at all a general failure but rather a selective breakdown. Standard grammatical categories (determiners, pronouns, punctuation) work perfectly, while dialectal features (phonological contractions, code-switches, regional vocabulary) fail consistently.

This pattern has important implications for how we understand model "knowledge." The Spanish model has not learned Spanish as such, but rather Standard Spanish as represented in formal written texts like Wikipedia and news articles. When it encounters variation from this standard, it does not recognize alternate valid forms but rather treats them as errors or unknown input. This encoding of prescriptivism has real-life consequences. Dominican speakers using NLP tools experience their language being marked as incorrect, reinforcing enshrined social hierarchies about whose "language" is legitimate.

Claim 3: Equal data does not produce equal outcomes.

The controlled experiment demonstrates that performance disparities persist independent of evaluation set size. This finding refutes a common explanation for low-resource language failures, that they simply need more data. While data availability certainly matters for training models, the architectural and methodological assumptions of current NLP create structural barriers that cannot be overcome by dataset creation alone.

High-resource languages benefit from transfer learning, where models pretrained on massive corpora learn representations that generalize to new tasks and domains. Low-resource languages cannot leverage these advantages because pretrained models either do not exist (Yoruba) or are trained on ill-suited data (Dominican Spanish). The neural architectures that work well for English may not be optimal for languages with different morphological, phonological, or syntactic properties. Also, evaluation metrics developed for English may not capture the relevant dimensions of performance for other languages.

These structural biases mean that achieving equity requires not just scaling up data collection but rethinking fundamental assumptions about model design, training procedures, and evaluation frameworks. True multilingual NLP would involve collaborative development with speaker communities, training approaches that do not assume English-like/Indo-European-like properties, and evaluation that attends to the specific phenomena relevant in each language.

7.2 Technical Challenges as Research Findings

The methodological difficulties encountered during this project are themselves significant findings. These three “failures” illustrate broader problems in NLP infrastructure:

Finding 1: Stanza's exclusion of Yoruba despite available UD data

This gap between resource creation and tool support demonstrates that language inclusion requires more than dataset publication. Even when gold-standard annotated corpora exist in standard formats, model development and maintenance require sustained institutional support. The Yoruba UD treebank exists but remains relatively unused because major NLP libraries do not provide pretrained models, creating a chicken-and-egg problem: researchers avoid working on languages without tool support, reinforcing the absence of tools.

Finding 2: spaCy tokenization incompatibility with Universal Dependencies

The tokenization mismatch between spaCy and UD highlights a methodological challenge in multilingual evaluation: pretrained models embed preprocessing decisions (e.g. tokenization, normalization, casing) that may be unclear to end-users and incompatible with standard benchmarks. This creates difficulties for fair comparison and reproducible research. While workarounds do exist, they introduce complexity and potential error sources that can handicap researchers working on less common languages that cannot rely on established pipelines.

Finding 3: Absence of annotated dialectal corpora

The lack of any existing annotated corpus for Dominican Spanish, despite its 13 million speakers (SIL International, 2015), forced the creation of a programmatically generated evaluation set. This highlights the systematic exclusion of dialectal varieties from corpus linguistics and NLP. While standard varieties are documented in large balanced corpora, dialects exist primarily in informal speech and social media, genres that are expensive to annotate and often excluded from academic corpora due to transcription complexity, stigmatization, etc.

These infrastructure gaps disproportionately affect researchers from underrepresented communities who may have native expertise in excluded languages but lack institutional resources to create training data, develop models, and maintain computational infrastructure. Addressing these inequities requires not just technical solutions but policy changes around research funding, dataset licensing, and institutional priorities.

7.3 Limitations

Several limitations affect the interpretation of these findings:

Corpus Size Disparity: The Dominican Spanish corpus (870 tokens) is substantially smaller than other test sets (8,000-25,000 tokens). While the controlled experiment addresses this concern by equalizing sizes across languages, the Dominican corpus remains smaller even in that condition. This means that error patterns might be less stable and some rare POS categories may not appear frequently enough to assess reliably.

Programmatic Generation: The Dominican corpus was generated using templates rather than collected from naturalistic sources. While I manually reviewed the corpus for authenticity, it still may not fully capture the complexity of real Dominican Spanish discourse. Patterns like discourse markers, topic fronting, and pragmatic functions that emerge in spontaneous speech or informal writing are underrepresented.

Written Language Bias: All corpora represent written language, even when transcribing spoken contractions. Features that exist primarily in pronunciation but not orthography cannot be

evaluated. This limitation affects all varieties but may be particularly significant for dialectal varieties where spoken and written forms diverge more substantially.

Single Tool Evaluation: Due to tokenization incompatibility, only Stanza was evaluated. Other tools like spaCy, Trankit, or multilingual transformers (mBERT, XLM-R) might show different error patterns. Stanza's LSTM architecture may have different encoded biases than transformer models. However, Stanza's direct training on UD data makes it the most appropriate tool for UD-based evaluation.

Limited Error Taxonomy: While I identified broad categories of dialectal errors (contractions, code-switches, lexical items), a more detailed and in-depth linguistic analysis could distinguish phonological, morphological, syntactic, and pragmatic sources of error. Deeper investigation of why specific tokens fail (lack of orthographic exposure, morphological ambiguity, distributional difference from standard) would provide more actionable insights.

Single Dialect: The Dominican Spanish corpus represents urban Santo Domingo usage and may not capture variation within the Dominican Republic or other Caribbean Spanish varieties. A comprehensive assessment would evaluate multiple dialects across the Spanish-speaking Caribbean, comparing Peninsular, Mexican, and other Caribbean varieties systematically.

7.4 Future Research Directions

This project opens several possible avenues for future investigation:

Direction 1: Training Yoruba models

The most immediate extension would be to train actual POS tagging models for Yoruba using the existing UD treebank. This would establish a true performance baseline rather than the naive tagger used here. Such models could be trained using Stanza's architecture or more recent approaches like multilingual transformers. Comparing Yoruba model performance to high-resource languages when trained on comparable amounts of data would provide cleaner evidence about linguistic versus infrastructure factors.

Direction 2: Expanding dialectal corpora

Creating larger, naturalistically sourced corpora for Dominican Spanish and other Caribbean varieties would enable more robust evaluation against the standard model. Potential sources include social media (Twitter, Reddit), subtitles of Dominican films and television, and interviews or oral histories. These corpora could be annotated using crowdsourcing approaches that engage native speakers, creating both evaluation data and training resources for potential dialect-adapted models.

Direction 3: Cross-linguistic dialect comparison

Extending this methodology to other languages with substantial dialectal variation (e.g., Arabic varieties) would test whether the patterns observed for Spanish generalize to other linguistic contexts. Are phonological contractions always problematic for standard-trained models? Does code-switching difficulty depend on the typological distance between switched languages? Do models fail more severely on stigmatized varieties versus regionally accepted standards?

Direction 4: Architectural interventions

Technical approaches to reducing dialectal bias might include: (a) training models on examples that include contracted and non-standard forms, so the model learns these are valid rather than errors, (b) teaching models to handle both standard and dialectal language simultaneously, treating them as related varieties rather than one being "correct," (c) starting with models trained on standard language and adjusting them using smaller samples of dialectal text, which requires less dialectal data, or (d) using models that analyze language at the character level rather than whole words, which helps them recognize spelling variations like *está* and *ta* as related forms. Evaluating these approaches would provide practical paths toward more equitable tools.

Direction 5: Multilingual model evaluation

Large multilingual transformers like mBERT and XLM-RoBERTa claim to learn cross-lingual representations that transfer across languages. Evaluating these models on the same language sample would test whether their multilingual pretraining mitigates some of the disparities observed with language-specific models. Do multilingual models perform better on low-resource languages by leveraging transfer from high-resource ones? Or do they simply reproduce the biases of their predominantly English training data?

8. Conclusion

This project demonstrates systematic inequities in NLP technologies across languages and dialects. Evaluating POS tagging performance on English, Standard Spanish, Yoruba, and Dominican Spanish reveals a 78 percentage point gap between highest- and lowest-performing varieties, with low-resource languages and non-standard dialects experiencing failures while high-resource standard varieties achieve near-human performance.

These disparities reflect not at all linguistic properties but infrastructure allocation. Yoruba, with 48 million L1 speakers and an available UD treebank, achieves only 18% accuracy because major NLP libraries provide no pretrained models. Dominican Spanish, a variety spoken by 13 million people, achieves 73% accuracy compared to Standard Spanish's 93% because models are trained on formal written registers that exclude dialectal features. The controlled experiment confirming that equal evaluation data does not produce equal outcomes demonstrates that these problems are structural, embedded in model architectures and training procedures that assume English-like properties and standard language norms.

Error analysis reveals that failures are not random but systematic. Standard grammatical categories (determiners, pronouns, punctuation) work reliably even in Dominican Spanish, while dialectal features (phonological contractions, code-switches, regional vocabulary) fail consistently. This selective breakdown indicates that models encode standard language ideology, treating variation as error rather than recognizing alternate valid forms.

The technical challenges encountered during this research: Stanza's exclusion of Yoruba, spaCy's incompatible tokenization, and the absence of annotated dialectal corpora are themselves findings that illuminate the structural barriers facing researchers working on non-European, non-standard varieties. These infrastructural gaps disproportionately affect scholars from underrepresented communities who possess native expertise in excluded languages but lack institutional resources for large-scale model development.

Achieving equitable language technologies requires more than dataset creation. It demands collaborative development with speaker communities, training approaches that do not assume English-like properties, evaluation that attends to language-specific phenomena, and institutional prioritization of linguistic diversity in research funding and NLP tool development. The alternative is a future that works amazingly for speakers of a handful of dominant standard varieties while systematically failing speakers of the world's thousands of other languages and dialects.

As someone deeply invested in both Dominican Spanish and Yoruba, I undertook this experiment to quantify what speakers of these languages and dialects already know by experience, that language technologies do not work for us. My hope is that experiments like mine, which measure and document these failures precisely, can contribute to advocacy for more equitable infrastructure and ultimately to NLP systems that recognize the linguistic diversity of human language, rather than encoding and enshrining the dominance of a privileged few.

References

Alba, O. (2004). *Cómo hablamos los dominicanos: Un enfoque sociolingüístico.* Fundación León Jimenes. https://fundacionleon.org.do/wp-content/uploads/2020/11/doc-comohablamos.pdf

Blodgett, S. L., Green, L., & O'Connor, B. (2016). Demographic Dialectal Variation in Social Media: A Case Study of African-American English. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1119–1130. https://doi.org/10.18653/v1/D16-1120

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., & Johnson, M. (2020). XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. *Proceedings of the 37th International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2003.11080

Ishola, Ọ., & Zeman D. (2020). Yorùbá Dependency Treebank (YTB). *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 5178-5186. https://aclanthology.org/2020.lrec-1.637/

Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The State and Fate of Linguistic Diversity and Inclusion in the NLP World. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6282–6293. https://doi.org/10.18653/v1/2020.acl-main.560

Jurgens, D., Tsvetkov, Y., & Jurafsky, D. (2017). Incorporating Dialectal Variability for Socially Equitable Language Identification. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 51-57. https://doi.org/10.18653/v1/P17-2009

Milroy, J. (2001). Language ideologies and the consequences of standardization. *Journal of Sociolinguistics*, 5: 530-555. https://doi.org/10.1111/1467-9481.00163

Nivre, J., de Marneffe, M-C., Ginter, F., et al. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, 1659-1666. https://aclanthology.org/L16-1262/

Orife, I., Kreutzer, J., Sibanda, B., et al. (2020). Masakhane -- Machine Translation For Africa. *AfricaNLP Workshop at ICLR 2020*. https://doi.org/10.48550/arXiv.2003.11529

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 101-108. https://doi.org/10.18653/v1/2020.acl-demos.14

SIL International (2015). Spanish Language (SPA). *Ethnologue: Languages of the world* (18th ed.). https://www.ethnologue.com/language/spa/ (subscription required)

SIL International (2025). Yoruba Language (YOR). *Ethnologue: Languages of the world* (28th ed.). https://www.ethnologue.com/language/yor/ (subscription required)