Analysis of LLM as a grammatical feature tagger for African American English

Anonymous ACL submission

Abstract

African American English (AAE) presents unique challenges in natural language processing (NLP) due to the limited availability of annotated data. This research systematically compares the performance of various NLP models—rule-based, transformer-based, and large language models (LLMs)—in identifying key grammatical features of AAE, namely Habitual Be and Multiple Negation. These features were selected for their distinct levels of grammatical complexity and occurrence. The evaluation involved sentence-level binary classification tasks, using both zero-shot and few-shot prompting strategies. The analysis revealed that while LLMs show promise, they are influenced by biases such as recency and formality, which impact their performance. The study highlights the necessity for improved model training and architectural adjustments to better accommodate AAE's linguistic characteristics. Data and code are available here.

1 Introduction

African American English (AAE) is considered a low-resource language, facing the challenge of inadequate annotated data for training natural language processing (NLP) models. While efforts have shown promise in improving NLP performance on AAE (Dacon, 2022; Masis et al., 2023), progress is limited. Consequently, AAE lacks access to the same host of language-specific NLP tools available to dominant varieties such as Mainstream American English (MAE). Therefore, NLP systems need to be capable of recognizing AAE usage.

This paper evaluates the ability of different NLP systems to recognize distinctive AAE grammatical features, comparing a rule-based model, a transformer-based model, and large language models (LLM) on the same tasks. It uses LLM zero- or few-shot classification of AAE, more specifically tagging distinctive AAE linguistic features. It uses two linguistic features as a test case: Habitual Be and Multiple Negation. Habitual Be is rare and difficult to tag while Multiple Negation is prevalent across various non-standard English dialects and can be captured easily using a pure rule-based models. Therefore, these two features provide extremity in terms of grammatical complexity and and possible prior exposure by LLMs.

2 Related Work

NLP research for AAE have explored dependency parsing (Blodgett et al., 2018), POS-tagging (Dacon, 2022; Jørgensen et al., 2016), hate speech classification (Harris et al., 2022; Sap et al., 2019), automatic speech recognition (Koenecke et al., 2020; Martin and Tang, 2020), dialectal analysis (Blodgett et al., 2016; Dacon, 2022; Stewart, 2014) and feature detection (Masis et al., 2022; Santiago et al., 2022; Previlon et al., 2024). Methods mitigating bias in NLP often neglect AAE's unique morphosyntactic features. Semantic context and lexical choice are more commonly accounted for (Barikeri et al., 2021; Cheng et al., 2022; Garimella et al., 2022; Hwang et al., 2020; Kiritchenko and Mohammad, 2018; Maronikolakis et al., 2022; Silva et al., 2021), but research often removes its morphological features (Tan et al., 2020) or translating between standard written English and AAE (Ziems et al., 2023). Our work highlights the lack of attention in NLP to AAE's distinctive grammatical features that can be used to identify its usage accurately.

3 African American Grammatical Features

Our work focuses on two grammatical features that characterize AAE and distinguish it from MAE.

3.1 Habitual Be

Although relatively infrequent, habitual *be* is employed regularly by AAE speakers (Blodgett et al., 2016) as well as speakers of 90 other Englishes (Kortmann, 2020). It an aspectual marker denoting a recurring, or habitual, action (Green, 2002; Fasold, 1969). In contrast to other instances of “be", the habitual *be* is invariant, meaning it never changes form. As shown in the example below, the MAE sentence requires the verb “be" to agree with “I" resulting in “I am" rather than “I be". Additionally, the adverb "usually" is indicates the event is recurring. AAE does not require “be” to change form nor does it require additional adverbs to express habituality.

> **AAE**: I be in my office by 7:30.
> **MAE**: ‘I am usually in my office by 7:30’

3.2 Multiple negation

Multiple negation, also called Double Negative or Negative Concord, is characterized by multiple negative words such that each negator confirms an overall negative meaning. Multiple negation was used in Old English Kallel (2011) and is today a recognized feature of several English varieties. However, it is absent from MAE as well as from most other modern Germanic languages Martínez (2003).

> **AAE**: I ain’t step on no dog.
> **MAE**: ‘I didn’t step on a dog.’

4 Data & Corpus

All data used for the training, testing, and analysis of the included models was transcribed, tagged, and annotated for AAE features by hand from entries in the African American History Project (AAHP), a part of the larger Samuel Proctor Oral History Program (SPOHP) (Moeller et al., 2024) or in The Corpus of Regional African American Language (CORAAL) (Kendall and Farrington, 2021).The materials are used in accordance with SPOHP’s guidelines for academic research and educational purposes. Proper credit is given to the program for the original data sources, and the use of these materials complies with the program’s policies on non-commercial, educational, and research use. This research uses data from the Corpus of Regional African American Language (CORAAL), which is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0). The dataset is used for non-commercial academic research, with proper attribution given to the creators. All usage complies with the terms outlined in the license, ensuring that the data is used for educational and research purposes only.Entries in this database are processed in a pipeline with each sentence being marked for the presence of AAE grammatical or phonological features.

Training/testing sets are generated for individual grammatical features. The features labels assigned by human annotators are treated as gold standard when analyzing the performance of the models. Datasets are created with a predetermined ratio of ‘positive examples’that contain words/phrases involved in the grammatical feature, such as the sentence “It used to be so hot.” as a negative example for Habitual Be. The task were set up as binary classification on sentences. The model tag the sentence as either positive or negative for the feature of interest.

5 Model architecture and configurations

We compared two LLMs, one transformer model, and one rule-based model. For the sentence-level binary classification tasks, each task involved analyzing, training, or prompting these models on batches of sentences, which were processed with a consistent prompt format, to classify test sentences based on a specific AAE feature.

5.1 Large Language Models

The language models used in this study were specifically chosen based on their suitability for sentence-level binary classification tasks, given the context and parameter constraints. Each model was configured with a consistent set of hyperparameters and prompt structures to enable a fair evaluation of their performance across various sentence structures.

OpenAI – GPT. OpenAI’s `gpt-4o-mini` model, a compact language model containing approximately 8 billion parameters, was accessed through the OpenAI REST API with a configuration set to a temperature of 0.7 and a top-p value of 0.9, providing a balance between response coherence and variability. The GPT model used in this research were accessed through OpenAI’s platform following OpenAI’s terms of service and usage policies

Meta – LLaMA Meta’s `LLaMA 3-8B-Instruct` has a similar architecture and parameter size (8 billion parameters) to OpenAI’s `gpt-4o-mini`. The `LLaMA` model was accessed through the

Huggingface transformers libraryexecuted using PyTorch v12.4 with NVIDIA CUDA support. The experimental settings, including temperature (0.7) and top-p (0.9), were set identical to those used for gpt-4o-mini to maintain consistency in response characteristics. The LLaMA models were used under the non-commercial research license provided by Meta, ensuring compliance with the restrictions specified in the LLaMA License Agreement. The models were used solely for academic research purposes and in accordance with ethical guidelines for artificial intelligence research.

5.2 Baselines

In order to assess the LLM results, we compared them to baseline feature taggers and models previously built. The Habitual Be baseline is a Transformer model based on the work in Previlon et al. (2024). The inputs to the model include n-grams windows around "be", predictions of feature based on part of speech (POS) windows and syntactic structures, as well as some post-hoc rules. The POS predictions were largely taken from Santiago et al. (2022), while the rest are found in Previlon et al. (2024). Since training data was limited, we used 10-fold cross validation and augmentated approximately 3,500 more Habitual Be sentences as described in (Santiago et al., 2022) to balance the dataset. The dataset contained on a total of 7,739 samples (4,009 non-habitual be and 3,730 habitual be). The data was split into 10 folds containing 774 samples (401 non-habitual and 373 habitual) each. For our results, we averaged performance across the 10 folds. Our data also contains augmented samples. We did this in order to be able to compare it like for like with the baseline transformer model used for habitual be and to get a sense of how the model would perform on average on any kind of data, real world or synthetic.

For the multiple negation tagger, we analyzed the patterns found in the data and wrote a rule-based tagger that predicted the presence of multiple negation or lack thereof in our data at nearly 100% accuracy. It works by identifying a negator (e.g. "not", "nothing") in a sentence and searching for a second instance within the clause. Clause boundaries are defined by an instance of punctuation or conjunctions. If a second instance was found, the sentence was tagged as '1', otherwise, '0'. The dataset contained a total of 198 samples (161 singular negation and 37 multiple negation). After running the classification report, only one of those sample was tagged incorrectly.

6 Prompt Design

In LLMs like GPT4o, prompt design plays a crucial role. Prompt length and structure can impact the responses generated, making it important to optimize prompt formulation for specific tasks. After comparing LLMs to the baselines, we analyze variations in prompt instructions.The simpler and more precise the instructions, the easier it is for the model to generate structured and predictable outputs, thereby reducing the need for post-processing.

The same input phrased differently can result in outputs of varying lengths and formats. For instance, we found a prompt like:

```
"Classify the sentence '{sentence}' as 'habitual
     be' or 'non-habitual be' in one word:"
```

results in the desired concise response. Without specifying the "in one word" constraint, a slightly altered prompt such as:

```
"Classify the sentence '{sentence}' as 'habitual
     be' or 'non-habitual be':"
```

often leads to the model adding explanations such as "This sentence uses 'be' in a habitual sense because it describes a repeated action," instead of providing just the classification label.

This behavior arises because the absence of strict word count or formatting instructions leaves the model room to interpret the output length and detail, which defaults to more verbose responses Even without explicitly requesting, the model may include justifications unless the prompt is explicitly constrained. Other prompts like:

```
"Determine if '{sentence}' is 'habitual be' or '
    non-habitual be'."}
```

which might yield an answer like, "This is a habitual use because the context suggests a repeated action," instead of just the classification term.

It is not enough to have an LLM available to use for a feature tagging. Prompt engineering reduces the need for complex post-processing, making the model more effective for real-world applications.

In our experiments, the prompt format for Zero Shot Prompting followed a template of:

```
"[batch index]. Classify the sentence '[Sample
    Sentence]' as '[Feature A]' or '[Feature B]'
     in one word while preserving the numbering
    at the start of the prompt."
```

The prompt format for Few Shot Prompting followed a template of:

1. Initiated with a predefined text: "I have given a few classified train examples," setting the context for the model.

2. Appended each selected training example to the prompt, formatted as:

```
"Sentence: [sentence from data]
Label: [corresponding label]"
```

3. Each example was clearly marked with its respective label, either 'habitual be' or 'non-habitual be', based on the 'Habituality' feature in the data.

The prompt generation function transitioned to constructing a set of classification tasks for the model.

7 LLM Compared to Baselines

We compared the each baseline model to the two LLM, using the classification report from `sklearn` (Pedregosa et al., 2011) to compare the model tags to the gold-standard tags. As compared to our habitual be baseline model, the few shot LLM models performed worse for both habitual be recall and overall F_1-score, by about -0.02 for the former and -0.18 for the latter. As compared to our multiple negation baseline tagger, the few short LLM model performed worse for both multiple negation recall and overall F_1-score, by about -0.08 for the former and -0.16 for the latter. In both comparisons, we noticed the LLM performs worse than the baseline, even with few shot learning. In order to understand the source of error within the LLM models, we tested two hypotheses – recency bias and formality bias.

7.1 Results of Baseline

The Habitual Be baseline model was evaluated on each of the 10 folds. The model returned, on average, a 0.88 recall and a 0.92 overall F_1-score on the habitual class.

The multiple negation baseline model achieved a recall of 1.00 and a 0.99 overall F_1-score. This means that all sentences containing multiple negation were correctly classified.

These are the results against which we evaluated the LLM models.

	Multiple Negation				**Habitual Be**			
	GPT		LLaMA		GPT		LLaMA	
	Zero	Few	Zero	Few	Zero	Few	Zero	Few
P_+	0.78	0.47	0.94	1.00	0.67	0.69	0.21	0.48
P_-	1.00	0.98	0.38	0.56	0.68	0.83	0.55	0.66
R_+	1.00	0.94	0.72	0.35	0.65	0.86	0.04	0.73
R_-	0.94	0.77	0.78	1.00	0.70	0.63	0.88	0.39
F_1w	0.95	0.82	0.76	0.59	0.67	0.74	0.41	0.53

Table 1: Classification performance of Multiple Negation and Habitual Be detection using the GPT model and the LLaMA model using zero-shot and few-shot prompting strategies. P_+ and P_-: precision of the positive class (feature present) and of the negative class (feature absent); R_+ and R_-: recall of the positive and negative classes; F_1w: weighted F_1

7.2 Results of Large Language Model

The LLM models results for zero shot and few shot prompting strategies for both multiple negation and habitual be features on the GPT and LLaMA models are shown in Table 1. In the table, the '+' class indicates presence of the feature and '-' indicates absence of the feature.

7.2.1 Habitual Be

For Habitual Be, the GPT model performance for zero-shot prompting was quite poor compared to few shot especially when we compare the numbers in terms of how many true habitual be samples the LLM was able to capture with both the prompting styles. This can be attributed to the complex and nuanced nature of Habitual Be. When the LLM looks at a few examples, it gets a slightly better understanding of what underlying syntactic and semantic structure to look for, to detect Habitual Be. The increase in performance of the GPT model is more so with the recall for the habitual class of the model while the precision stays almost the same. This may be because with prompting, the model may be biased more towards detecting that a sentence contains a habitual be feature.

The performance with the LLaMA model could not be computed on all the 7,739 samples due to memory constraints and rate limits. So, we performed the experiment with a reduced total dataset of 1,000 samples (430 labeled Habitual and 570 labelled Non-Habitual). We utilized a similar 10 fold data split as the GPT model but because the number of samples for each fold was lower, we decided to generate a combined classification report for all the 1000 samples. The LLaMA model is

unable to detect habitual be samples well without being nudged by a few examples using few shot.

7.2.2 Multiple Negation

For the multiple negation model, we tested the performance of the GPT model on a dataset containing 198 samples, out of which 162 were labeled Multiple Negation and the remaining 36 were labeled Non-Multiple Negation. The zero-shot prompting produced better results compared to few shot on GPT. This performance difference is prominent when we look at the dip in precision scores for both the positive (multiple negation) and negative (not multiple negation) class. The high recall and low precision of the LLM is because the LLMs are biased towards predicting any sentence as multiple negation if there are two or more negative words in a sentence. Multiple Negation is a relatively simpler feature to detect, but it is also easy to differentiate it based on the number of negative words the LLM sees in a sentence and this is one of the reasons the model tends to have a higher recall and lower precision. The precision score falls down further with few shot prompting likely due to being overfit to the few extra samples that it learns in the prompt. For a simple to detect feature like multiple negation, it may be counter productive to feed the LLM more examples and add more bias to it.

We also tested the performance of `LLaMA` model for multiple negation with the same dataset of 198 samples. The `LLaMA` model performed poor compared to GPT overall, but its precision for the positive class is much higher than the GPT model. But this appears to be more due to the model's bias towards predicting that the sentences do not contain the multiple negation feature. The lower recall for `LLaMA` can be attributed to it being trained on much smaller data compared to GPT. We notice a similar trend with `LLaMA` for multiple negation that we saw with gpt for multiple negation that performance with zero shot is better than with few shot.

Overall, the GPT model performed better than `LLaMA` both in terms of throughput as well as quality of detection, compared to the `LLaMA` model. It is evident that the `LLaMA` model has not been trained on a lot of data from African American English. The increase in performance by showing it a few examples with few shot helped the model immensely. This shows that fine tuning the model with more data from African American English could improve it's performance. The performance of the GPT model was quite contrasting for the habitual be and multiple negation feature detection. While the model performed poorly on habitual be, especially for zeroshot, the performance on multiple negation was much better. This can be attributed to the syntactic structural complexity of the 2 features. Multiple Negation is relatively simpler than habitual be and hence, LLMs do not need any kind of examples to be able to detect it. Habitual Be, on the other hand, is a bit more complicated. There is a lot of variability in its usage in African American English and without having the context or extra data regarding this, LLMs find it harder to detect it. Overall, for both the features, LLMs seem to be biased towards predicting the presence of a feature.

8 Potential LLM Biases

Upon further inspection of the predictions to understand the cause for the bias towards predicting the presence of a feature, we saw four major factors influencing the predictions from the LLMs and took them up for further analysis to quantify the overall impact of each of them on the models.

Recency bias - Recency bias means that the LLM's predictions for the current sentence can be affected by the previous sentence's prediction.As the input to the LLMs in our experiment were in batches, recency bias from the previous sentences and predictions in the same batch can influence the model's predictions. We observed in some sentences with wrong predictions, where the model predicted that the sentence contained a multiple negation feature when it did not.

Formality bias - Run-on, incomplete and disjoint sentences with excessive punctuation or fragmentation. While LLMs performed well on most sentences with some punctuation, we observed that for longer sentences and disjoint sentences, where there is a higher number of punctuations like semicolon and comma, the performance degraded. Even in the case of sentences containing grammatical errors, the model tends to perform poorly. This could be because LLMs are trained on written English and our work is based on transcripts of spoken English, so the grammatical deviations of spoken English permeate to the transcripts and so LLMs may have a hard time with such sentences.

AAE feature bias - Presence of other African American English features. We observed many examples where the model predicted presence of multiple negation and habitual be when there was presence of a different African American English

features like Person Number disagreement or the use of the word 'ain't'. This could be a major reason for the models predicting that a sentence contains a feature, when they don't. As shown in the examples below, LLMs predicted these sentences as containing multiple negation, but they do not. They do contain another AAE features like person number disagreement as can be seen in the first sentence in the phrase "We was". In Standard American English, it would've been written as "We were". In the second example, "ain't" is the other AAE feature. Based on these examples, we believe that LLMs can be biased towards detecting the presence of an AAE feature based on the presence of a different AAE feature.

> **Example 1**: We was in Pentecost holiness and I wasn't allowed to smoke.

> **Example 2**: Because he ain't been back to finish yet.

We decided to systematically examine the Recency bias and Formality bias factors further to verify if these factors do influence LLMs and to quantify their impact on the LLMs performance for the task of detecting multiple negation and habitual be features.

8.1 Hypothesis 1: Recency Bias

8.1.1 Methodology

Recency bias in the LLMs' predictions was assessed using logistic regression fitted using Maximum Likelihood Estimation on aforementioned experiments, where the relevant input sentences were presented in a relatively random order without probing for any specific error. The regression was conducted using the Python packages 'statsmodels', 'numpy', and 'pandas'. The dependent variable is the model prediction (presence or absent of a feature). The independent variable of interest is the presence of a recency bias. It is operationalised as the proportion of the predicted values of the last five samples (N_{-5},...N_{-1}) that match the predicted value of the given sample (N). The recency bias can either positively or negatively affect the prediction. A positive effect indicates the model prefers using recently predicted values, while a negative effect indicates the model avoids using recently predicted values. Ground truths were treated as an additional independent variable to control for the expected performance of the model, since the predicted values are expected to positively correlate with the ground truth.

8.1.2 Results: Zero-shot Prompting

	Multiple Negation		**Habitual Be**	
	GPT	LLaMA	GPT	LLaMA
Recency $\hat{\beta}$	-8.34	-4.87	-0.21	-16.52
Recency p	<0.001	<0.001	0.42	<0.001
Ground $\hat{\beta}$	5.59	2.17	1.70	1.48
Ground p	<0.001	<0.001	<0.001	0.15
Pseudo R^2	0.77	0.28	0.12	0.90

Table 2: Recency bias – Logistics regression summaries of Multiple Negation and Habitual Be detection using GPT and LLaMA models with zero-shot prompting. Recency/Ground $\hat{\beta}$ and Recency/Ground p denote the coefficient and the p-value of the recency bias variable and the ground truth variable.

8.1.3 Analysis: Zero-shot Prompting

The logistic regression analysis reveals whether the recency bias has an effect on sentence classification of two AAE features under each prompting paradigm. The regression summaries can be found in Table 3.

Habitual Be detection with GPT The regression analysis reveal that the $\hat{\beta}$ for recency bias to be -0.21 and it is not statistically significant (p-value = 0.42). This indicates that, in the zero-shot setting, recency bias does not have a meaningful impact on the model's ability to classify Habitual Be. Unsurprisingly, the control variable (ground truths) has a positive effect on ($\hat{\beta}$ = 1.740, p-value <0.001).

Multiple Negation detection with GPT For the zero-shot scenario for Multiple Negation feature in the GPT Model, the results indicate a strong *negative* relationship between recency bias and model performance ($\hat{\beta}$ = -8.34, p-value <0.001). This suggests that the model avoids generating a predicted value that it recently used. Unsurprisingly, the control variable (ground truths) has a positive effect on ($\hat{\beta}$ = 5.59, p-value <0.001).

Multiple Negation & Habitual Be detection with LLaMA Recency bias was found also with both multiple negation and Habitual Be detection using LLaMA. Similar to the GPT model, there is a strong *negative* relationship between recency bias and the LLaMA model's performance to detect multiple negation ($\hat{\beta}$ = -4.88, p-value <0.001). While the GPT model does not have a recency bias when

detecting habitual be, the LLaMA model has such a bias ($\hat{\beta}$ = -16.52, p-value <0.001).

With the exception of the GPT model that detects habitual be, the recency bias is stronger with habitual be than with multiple negation. The exact nature of this difference might be due to the fact that habitual be has a range of complex syntactic environments (Previlon et al., 2024), while multiple negation is easier to describe. Therefore, multiple negation detection could be somewhat self-evident without providing an example. On the one hand, LLaMA would typically mark any sentence that contained more than one negation of any kind as being Multiple Negation, with this often being a correct assumption. On the other hand, the Habitual Be feature gives little away by name, relying on more experience interpreting non-standard American English to recognize without additional examples. Given that the zero-shot experiments do not provide explicit examples, one can speculate that LLaMA is using previous inputs and their predictions to guide future prediction due to a lack of sufficient training on AAE grammatical features in the training data of the LLM (Martin, 2022).

8.1.4 Additional analysis: Few-Shot prompting

Few-shot prompting (GPT, Habitual Be) In contrast, the few-shot prompting results indicate a significant negative impact of recency bias on prediction accuracy ($\hat{\beta}$ = -1.03, p-value = 0.003). This substantial effect suggests that the few-shot model is considerably influenced by the timing and sequence of input data, potentially leading to biased predictions based on recent but not necessarily representative samples. The true label in this scenario also exhibits a strong positive influence ($\hat{\beta}$ = 3.42, p-value < 0.001), reinforcing the label's dominant role in driving predictions. With a higher Pseudo R^2 of 0.33, the few-shot model demonstrates a better fit than the zero-shot model, indicating a more robust explanation of the prediction dynamics.

Few-shot prompting (GPT, Multiple Negation) In the few-shot prompting scenario for the Multiple Negation Feature in the GPT Model, the analysis presents an even more pronounced negative impact of recency bias ($\hat{\beta}$ = -12.35). This stronger negative effect could be attributed to the few-shot model's reliance on a limited number of examples, which may accentuate the influence of recent inputs, thereby skewing prediction outcomes significantly. The presence of a lower, yet still positive, coefficient for the ground truth variable ($\hat{\beta}$ = 4.13) underscores a consistent trend where the intrinsic attributes of labels influence model predictions, but the effect of recency bias remains a dominant factor, adversely affecting prediction accuracy.

Some additional testing was done to more closely examine the relationship between LLaMA's particularly strong recency bias and the examples provided in few-shot testing. A set of ten prompts were chosen that the LLM failed on previous experiments, five with true labels '0' and five with true labels '1'. First, these were presented in alternating order, second with all consecutive 1-labeled sentences followed by all 0-labeled sentences, and finally with this ordering reversed. These two experiments were then run again with the batch size increased from 20 to 30. Each of these four tests were run six separate times, each one having the same set of examples ordered differently to measure if the order of provided samples impacted recency bias. Overall, few-shot prompting demonstrated substantially more sensitivity to recency in provided examples than sentence history, while still being much more accurate than zero-shot counterparts even when negatively impacted. The ordering of example sentences had a substantial impact on prediction accuracy while the ordering of input sentences largely did not, further suggesting that text within the most recently processed five to ten sentences have a strong impact on LLaMA's predictions when the criteria for a given feature's presence is not clear.

8.2 Hypothesis 2: Formality bias

Our second hypothesis was that the LLM models are hyper-sensitive to deviations from formal written texts with the according sentence structure and style. Therefore, they flag excessive punctuation, missing subjects or run on sentences as AAE features.

8.2.1 Methodology

Similar to Section 8.1.1, we employed a logistic regression analysis. The dependent variable is the predicted value of the models. To examine the hypothesis, the variable of interest is the presence of a deviation from formal written text. Each sentence was manually tagged as '1' if it contained excessive punctuation, missing subjects, or run on sentences, and as '0' if it did not. Ground truths were treated as a control variable. Due to the size of the habitual

be dataset, only fold 2 was tested in this hypothesis. We believe it is a fair representation of the entire Habitual Be dataset as it produced the median k-fold results of the zero-shot GPT model.

8.2.2 Results

	Multiple Negation		Habitual Be	
	Zero	Few	Zero	Few
Formality $\hat{\beta}$	0.17	0.89	0.10	0.96
Formality p	0.78	0.02	0.53	<0.001
Ground $\hat{\beta}$	5.39	3.67	1.42	1.77
Ground p	<0.001	<0.001	<0.001	<0.001
Pseudo R^2	0.54	0.26	0.09	0.17

Table 3: Formality bias – Logistics regression summaries of Multiple Negation and Habitual Be detection using GPT with zero-shot and few-shot prompting strategies. Formality/Ground $\hat{\beta}$ and Formality/Ground p denote the coefficient and the p-value of the formality bias variable and the ground truth variable.

8.2.3 Analysis

The logistic regression summaries can be found in Table 3. Unsurprisingly, the ground truth variable has a positive and significant effect in all four models. Turning to the bias variable, on the one hand, the GPT model with zero-shot prompting does not suffer from a formality bias with both features. While the formality bias variable has a positive $\hat{\beta}$ of 0.17 and 0.10 for Multiple Negation and Habitual Be respectively, it was statistically insignificant in both cases (ps > 0.05). On the other hand, the GPT model with few-shot prompting does suffer from such a bias with both features. Therefore, across the features, the bias variable has a significant effect only for the few-shot strategy as opposed to the zero-shot. This suggests that the influence of formality bias depends on the examples being given to the model.

The bias has a stronger effect on Habitual Be ($\hat{\beta}$ = 0.96, p <0.001) than on Multiple Negation ($\hat{\beta}$ = 0.86, p <0.001). This pattern of Habitual Be being more susceptible to the bias was also found in Section 8.1 with the recency bias. The positive effect of the formality bias, indicated by the positive $\hat{\beta}$s, suggests that, as predicted by our hypothesis, a sentence that deviates from standard written text (excessive punctuation, missing subjects, or run on sentence) increases the likelihood of the GPT model to predict it to have an AAE feature.

The few-shot prompting strategy should not be abandoned just because it suffers from the formality bias, since it was shown to enhance model performance with a complex feature such as Habitual Be (see Section 7.2.1). This warrants future work on mitigating this bias by experimenting with the nature and the proportion of the few-shot examples that deviate from standard written text.

9 Conclusion

Large Language Models (LLMs) continue to show promise in automating feature annotation tasks, yet significant challenges remain, particularly when detecting complex grammatical features such as the "Habitual Be" in African American English (AAE). The choice between employing zero-shot and few-shot approaches for annotation is heavily influenced by the complexity of the feature and the extent to which relevant data is represented in the LLM's pre-training corpus. While our baseline transformer model currently outperforms LLMs in detecting the Habitual Be, there are several strategies that can potentially enhance LLM performance, especially in light of the hypotheses we have explored. One promising approach is to mitigate recency bias, a known issue in LLMs, by interleaving or randomizing the order of sentences presented for prediction. This technique significantly reduces the model's dependency on the sequence of inputs, which can otherwise skew results. Additionally, pre-processing the input data by removing extraneous punctuation and segmenting disjointed sentences may help minimize noise and errors introduced during annotation. The findings from this research provide valuable insights into the potential and limitations of LLMs for feature annotation in AAE. They underscore the need for further experimentation to assess LLM performance on other linguistic features specific to AAE. Additionally, exploring how the presence of multiple AAE features within a sentence influences the accuracy of LLM predictions for individual features could offer a more deeper understanding and lead to more robust annotation techniques.

Limitations

We only gathered transcribed spoken speech data for this experiment. We could have looked at more sources like written texts and speeches from regions other than Florida. Due to the time limitations, we were only able to cover 2 African American English features for this paper. The data with

which we have tested the performance of our benchmark rule based model and LLMs is limited. There is a possibility that the rule based model could deteriorate in performance with the presence of more data and more variability in the data, requiring more rules to capture different instances of multiple negation. Our regression analysis for recency bias considers the most recent 5 examples. We encoded the feature for this parameter as the proportion of samples in the last 5 predictions that match the current prediction and ran the regression analysis. What this misses is that, for example, where all the past 5 samples match the prediction, and the prediction is correct, there may be a very high coefficient for the recency bias, but we cannot say for certain if it is in fact due to recency bias or due to the LLM correctly predicting the feature. Fortunately, this problem only presented itself when we were experimenting with an extreme ordering of the recency bias examples, such as where all 1s and 0s were grouped together. As such, this is a highly unlikely case. Another limitation we faced was in testing hypothesis 2, due to the augmentation of the habitual be data. The lack of natural sentences structures and meanings made it more difficult to distinguish whether the sentences contained missing subjects or run on aspects. Finally, the last limitation we identify is of the comparison between the LLMs and transformers. The transformers performance may be explained by the fact that they have been trained explicitly for detecting the habitual be feature, while LLMs have not. But then, the goal of this paper was to test the capability of general purpose LLMs to tag AAE features without having to fine-tune the model.

Ethical considerations

The involved university does not require IRB approval for this kind of study, which uses publicly available data.

We do not see any other concrete risks concerning dual use of our research results. Of course, in the long run, any research results on AI methods based on large language models could potentially be used in contexts of harmful and unsafe applications of AI. It is possible that feature tagging of a language of specific group of people could be used to identify the demographics of the individuals who use African American English.

Contribution statement

[ANONYMIZED]

Acknowledgments

[ANONYMIZED]

References

Soumya Barikeri, Anne Lauscher, Ivan Vulic, and Goran Glavas. 2021. Redditbias: A real-world resource for bias evaluation and debiasing of conversational language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1941–1955. Association for Computational Linguistics.

Su Lin Blodgett, Lisa Green, and Brendan T. O'Connor. 2016. Demographic dialectal variation in social media: A case study of african-american english. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1119–1130. The Association for Computational Linguistics.

Su Lin Blodgett, Johnny Wei, and Brendan O'Connor. 2018. Twitter Universal Dependency parsing for African-American and mainstream American English. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Melbourne, Australia. Association for Computational Linguistics.

Lu Cheng, Nayoung Kim, and Huan Liu. 2022. Debiasing word embeddings with nonlinear geometry. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1286–1298. International Committee on Computational Linguistics.

Jamell Dacon. 2022. Towards a deep multi-layered dialectal language analysis: A case study of African-American English. *Proceedings of the Second Workshop on Bridging Human–Computer Interaction and Natural Language Processing*.

Ralph W. Fasold. 1969. Tense and the Form Be in Black English. *Language*, 45(4):763–776. Publisher: Linguistic Society of America.

Aparna Garimella, Rada Mihalcea, and Akhash Amarnath. 2022. Demographic-aware language model fine-tuning as a bias mitigation technique. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP*

2022 - Volume 2: Short Papers, Online only, November 20-23, 2022, pages 311–319. Association for Computational Linguistics.

Lisa J. Green. 2002. *African American English: A Linguistic Introduction*. Cambridge University Press.

Camille Harris, Matan Halevy, Ayanna Howard, Amy Bruckman, and Diyi Yang. 2022. Exploring the role of grammar and word choice in bias toward African American English (AAE) in hate speech classification. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 789–798, New York, NY, USA. Association for Computing Machinery.

Alyssa Hwang, William R. Frey, and Kathleen McKeown. 2020. Towards augmenting lexical resources for slang and African American English. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 160–172, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Anna Jørgensen, Dirk Hovy, and Anders Søgaard. 2016. Learning a POS tagger for aave-like language. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1115–1120. The Association for Computational Linguistics.

Amel Kallel. 2011. The loss of negative concord in standard english: A case of lexical reanalysis.

Tyler Kendall and Charlie Farrington. 2021. The Corpus of Regional African American Language. Publisher: Online Resources for African American Language Project.

Svetlana Kiritchenko and Saif M. Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 43–53. Association for Computational Linguistics.

Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R Rickford, Dan Jurafsky, and Sharad Goel. 2020. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117(14):7684–7689.

Bernd Kortmann. 2020. *Syntactic Variation in English*, chapter 16. John Wiley & Sons, Ltd.

Antonis Maronikolakis, Philip Baader, and Hinrich Schütze. 2022. Analyzing hate speech data along racial, gender and intersectional axes. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 1–7, Seattle, Washington. Association for Computational Linguistics.

Joshua L. Martin. 2022. *Automatic Speech Recognition Systems, Spoken Corpora, and African American Language: An Examination of Linguistic Bias and Morphosyntactic Features*. Ph.D. thesis, University of Florida.

Joshua L. Martin and Kevin Tang. 2020. Understanding Racial Disparities in Automatic Speech Recognition: The Case of Habitual "be". In *Proc. Interspeech 2020*, pages 626–630.

Ignacio M. Palacios Martínez. 2003. Multiple negation in modern english. a preliminary corpus-based study. *Neuphilologische Mitteilungen*, 104(4):477–498.

Tessa Masis, Chloe Eggleston, Lisa J Green, Taylor Jones, Meghan Armstrong, and Brendan O'Connor. 2023. Investigating morphosyntactic variation in African American English on Twitter. *Proceedings of the Society for Computation in Linguistics*, 6(1):392–393.

Tessa Masis, Anissa Neal, Lisa Green, and Brendan O'Connor. 2022. Corpus-guided contrast sets for morphosyntactic feature detection in low-resource English varieties. In *Proceedings of the first workshop on NLP applications to field linguistics*, pages 11–25, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Sarah Moeller, Alexis Davis, Wilermine Previlon, Michael Bottini, and Kevin Tang. 2024. Compiling a corpus of African American language from oral histories. *Research in Corpus Linguistics*, 12(2):45–79.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Wilermine Previlon, Alice Rozet, Jotsna Gowda, Bill Dyer, Kevin Tang, and Sarah Moeller. 2024. Leveraging syntactic dependencies in disambiguation: The case of African American English. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10403–10415, Torino, Italia. ELRA and ICCL.

Harrison Santiago, Joshua Martin, Sarah Moeller, and Kevin Tang. 2022. Disambiguation of morphosyntactic features of African American English – the case of habitual be. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 70–75, Dublin, Ireland. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1668–1678. Association for Computational Linguistics.

Andrew Silva, Pradyumna Tambwekar, and Matthew Gombolay. 2021. Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2383–2389, Online. Association for Computational Linguistics.

Ian Stewart. 2014. Now we stronger than ever: African-American English syntax in Twitter. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 31–37. The Association for Computer Linguistics.

Samson Tan, Shafiq Joty, Lav Varshney, and Min-Yen Kan. 2020. Mind your inflections! Improving NLP for non-standard Englishes with Base-Inflection Encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5647–5663, Online. Association for Computational Linguistics.

Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta, and Diyi Yang. 2023. Multi-VALUE: A framework for cross-dialectal English NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 744–768, Toronto, Canada. Association for Computational Linguistics.