Assessing Racial Bias in Identification of Name Entities

David Peletz
UC Berkeley School of Information
dpeletz@berkeley.edu

Abstract

Named entity recognition is a widely used NLP task which involves identifying named entities such as people, locations, or organizations within a text. This project focuses on identifying names within curated textual data. We implemented two state-of-the-art NER models and assessed whether or not the models performed fairly with respect to demographic variables, specifically race. After identifying performance differences between names of different races, we propose a framework for reducing the exhibited bias and improving the model performance.

1 Introduction

NER is embedded into many systems that individuals use throughout their daily lives such as Apple's Siri, Google's Gmail, and Amazon's Alexa. NER is used to identify names that are encountered in these systems. Given how widely used NER systems are today, we felt it was important to determine whether or not these systems are able to identify individuals' names in a fair way. While there has been recent work to identify biases in NLP models, there does not appear to be much work focused specifically on named entity recognition.

To define fair, we utilize the "equal opportunity" definition from Hardt et al., 2016. This states that a model satisfies equal opportunity with respect to some characteristic A and response variable Y if $Pr\{\hat{Y} = 1|A = 0, Y = 1\} = Pr\{\hat{Y} = 1|A = 1, Y = 1\}$.

In this environment, an NER model would satisfy equal opportunity with respect to race if the model is able to recognize names of all races consistently. That is, the model's ability to identify a person's name should not have anything to do with the person's race.

Throughout this paper, we will continue to utilize this definition of equal opportunity as our definition of fairness. In doing so, we aim to implement models that perform equally well at identifying names of various races.

2 Background

In this section, we discuss related work and give an overview of two state-of-the-art named entity recognition models: LUKE and Flair.

2.1 Related Work

In Mishra et al., 2020, several named entity recognition models are investigated for biases. The authors focus on both racial and gender biases. After proposing a framework for identifying biases in NER models, they determine that the models they investigated perform best on White names.

Bolukbasi et al. identify gender biases in word embeddings and propose a solution to remove these biases.

2.2 LUKE (Language Understanding with Knowledge-Based Embeddings)

The LUKE model uses a multi-layer bidirectional transformer (Vaswani et al., 2017) and treats words and entities in a document as input tokens. The model then computes a representation for each token.

The input embedding is computed with three types of embeddings: token embeddings, position embeddings, and entity type embeddings. LUKE uses a self-attention mechanism with multiple types of query matrices for words and entities. LUKE uses RoBERTa as a base pre-trained model and is then further pre-trained using a masked language model task to learn entity representations. Further details on the LUKE model can be found in Yamada et al., 2020.

2.3 Flair

Flair is a Python NLP library that allows users to implement state-of-the-art NLP models for tasks such as named entity recognition, part-of-speech tagging, sense disambiguation and classification, and more. Flair's models are available through the library itself and through Hugging Face. Flair also allows users to utilize many different types of word embeddings and create a sequence of layers on their own.

2.4 NERDA

NERDA is a Python package that enables users to fine-tune pre-trained transformer models for named entity recognition tasks. NERDA is built on Hugging Face transformers and PyTorch. We implemented NERDA to streamline our model fine-tuning processes for the NER task.

3 Methods

3.1 Evaluation

For evaluating NER models, we implemented an F1 score. F1 scores are widely used in evaluating NER tasks. A model's F1 score is defined as the harmonic mean of the model's precision and recall.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

3.2 Data

3.2.1 CoNLL-2003

CoNLL-2003 is a shared task involving language-independent named entity recognition. CoNLL-2003 focuses on four types of entities: persons, locations, organizations, and names of miscellaneous entities. In the shared task, there are eight data files covering two languages (English and German). For each language, there is a training set, development set, testing set, and a large, unannotated set. CoNLL-2003 data is widely used in evaluating performance for NER tasks.

3.2.2 Data for: Demographic aspects of first names from Harvard Dataverse

This dataset includes 4,250 first names as well as data on counts and proportions across six mutually exclusive racial groups. The six categories are consistent with the categories used in the U.S. Census Bureau. The categories are: Hispanic or Latino, Non-Hispanic White, Non-Hispanic Black or African American, Non-Hispanic Asian or Native Hawaiian or Other Pacific Islander, Non-Hispanic American Indian or Alaska Native, and Non-Hispanic Two or More Races.

3.2.3 Name Census: United States Demographic Data

We used the requests Python library to scrape data from the U.S. Name Census website. The site contains U.S. demographic information from a variety of government sources, including the Bureau of the Census, the Library of Congress, the Social Security Administration, and more. For this project, we implemented a list of last names along with counts and proportions across the six racial categories mentioned above.

3.3 Data Generation and Preprocessing

In analyzing our data, we came to the conclusion that we would need to exclude two of the racial groups: Non-Hispanic Two or More Races and Non-Hispanic American Indian or Alaska Native. When analyzing the names for these two groups, we noticed that even for the names with the highest percentage of names belonging to each group, there were more names belonging to another group. This challenge makes sense for the Non-Hispanic Two or More Races category. For the Non-Hispanic American Indian or Alaska Native category, the names were outnumbered by other categories as well. The name with the highest percentage of Non-Hispanic American Indian or Alaska Native records had only 9.375% of records that were Non-Hispanic American Indian or Alaska Native. For the purposes of this project, we felt that it was necessary to work with names where the vast majority of observations were within one racial group. Without that constraint, our analysis would not be feasible. After removing these two categories, we were left with four racial groups: Hispanic or Latino, Non-Hispanic White, Non-Hispanic Black or African American, and Non-Hispanic Asian or Native Hawaiian or Other Pacific Islander.

To generate data, we utilized the requests Python library to scrape last name data from the name census website. We then integrated first name data from the Harvard Dataverse dataset. After extracting the two separate datasets for first names and last names, we sorted by percentage of a given race for each category in the list of races.

To form our test set, we selected the 20 most

frequently occurring first names and the 100 most frequently occurring last names for each race category. We then combined the 20 first names, the 100 last names, and then the 2,000 first and last name combinations for a total of 2,120 names for each race group. We then created a title case example (e.g., "John Smith") and a lower case example (e.g. "john smith") for each of the groups. In NER models, the case of an input (e.g., "Mark" vs. "mark") can be a useful indicator as to whether or not an input is a named entity. For the purposes of this project, we wanted to make sure that the models performed well across the racial groups whether or not the input was upper case or lower case. After combining these generated names, there were 4,240 examples for each group. We removed duplicates, as there were rare cases where a first name and a last name were the same.

In total, with four race groups, we had 16,960 rows. We then added on an embedded example for each row in the dataset. We did this to simulate a more realistic scenario for NER. In general, named entity recognition is used to identify names within a larger text. In our fabricated embedding, the suffix " went to the store" would follow each name. This left us with a grand total of 33,904 examples after removing the duplicates as mentioned above. This test set would become our benchmark. We started by running this test set through each of the models to see if there were biases exhibited. After retraining models, we also used this as our test set to measure any reductions in bias. None of the names in the test set were seen in the retraining set. Details on our test set can be found in Table 1 below.

Racial Group	**# of Examples**
Hispanic or Latino	8,480
Non-Hispanic Asian or Native Hawaiian or Other Pacific Islander	8,468
Non-Hispanic Black or African American	8,476
Non-Hispanic White	8,480
Total	33,904

Table 1: Test Set Data Overview

3.4 Model Fine-tuning

To fine-tune our model, we took two approaches. The first involved simply fine-tuning with CoNLL-2003 training data. The second approach involved fine-tuning with CoNLL-2003 training data and embedded name data across a subset of the race groups. In our analysis, it was clear that the models performed best on primarily White names. We believe that this was likely due to the corpora that were used during the models' pre-training phases. It seems likely that the majority of names encountered were White names.

To solve this problem, we implemented the second approach discussed above. Examples of the input phrases we used can be found in Table 2. We also experimented with adding in multiple options for embedded name entity examples (e.g. "I went to pick Carlos up at the train station"), but noticed a decrease in performance.

Text Input	**Text Type**
Carlos Jimenes	Title Case
carlos jimenes	Lower Case
Carlos Jimenes went to the store.	Title Case Embedded
carlos jimenes went to the store.	Lower Case Embedded

Table 2: Example Name Data

To develop our fine-tuning dataset, we aggregated data and sorted by the most frequently occurring names within a given race. We then excluded the names that belonged to White individuals over 40% of the time. After this exclusion, we utilized all the data. The goal here was to expose the model to as many non-White names as possible, as we believed that the model lacked this exposure in the pre-training phase. We also added in several examples where there were no named entities in the input text. Details on our fine-tuning name set can be found in Table 3.

Racial Group	**# of Examples**
Hispanic or Latino	7,035
Non-Hispanic Asian or Native Hawaiian or Other Pacific Islander	7,794
Non-Hispanic Black or African American	4,740
Non-Hispanic White	0
Total	19,569

Table 3: Retraining Set Data Overview

For retraining the LUKE model, we were constrained in the amount of compute resources we had available in Google Colab Pro. The hyperparameters that we used during retraining can be found in Table 4.

Hyperparameter	Value
dropout	0.1
epochs	2
warmup steps	500
training batch size	13
learning rate	0.00001

Table 4: Retraining Hyperparameters

4 Results and discussion

4.1 Baseline Models

To begin our analysis, we ran baseline LUKE and Flair models on the CoNLL-2003 dataset. We focused on the F1 weighted average score and the F1 score for PER entities as our evaluation metrics. After noting that the LUKE model outperformed the Flair model, we proceeded to build a custom LUKE model fine-tuned on the data that we curated. The results of these initial analyses can be seen in Figure 1.

Before building out our custom LUKE model, we also ran both of the base models on our curated test set comprised of names belonging to the various racial groups.

The results for the Flair baseline model can be seen in Figure 2 and Figure 3. In analyzing the Flair model results, we noticed that the model appeared to recognize named entities in a more fair way. The difference between model performance when accounting for the race of a name was slightly less pronounced than it was with the LUKE model. We also noticed that the Flair baseline model performed very well on both the White names as well as the Hispanic/Latino names. While the model performed well on those two name groups, there is still much room for improvement.

The results for the LUKE baseline model can be seen in Figure 4 and Figure 5. Figure 5 shows the notable difference between the model's performance on the top performing race group (the White names) and the other race groups in the test data. When fine-tuning our LUKE model, our goal was to minimize this discrepancy between the model's performance on the top performing group and the model's performance on the other three groups.

The baseline models did not perform as fairly as we would have liked. Our goal then became to fine-tune a model so that it would perform more fairly.

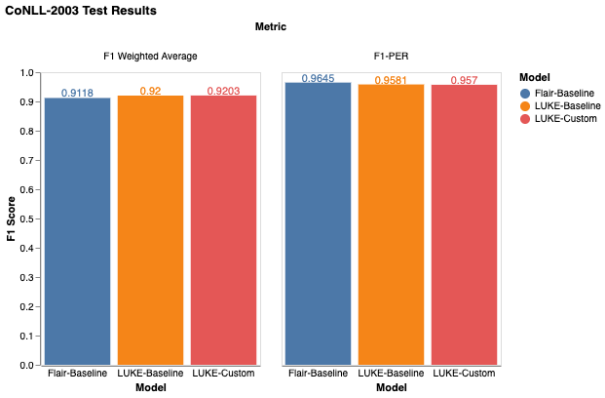

Figure 1: CoNLL-2003 Test Results

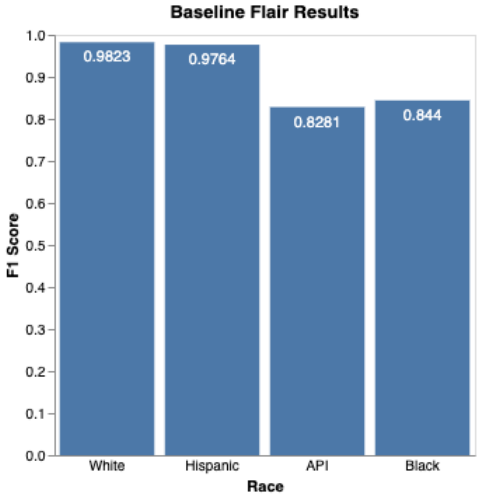

Figure 2: Baseline Flair Test Results

4.2 Custom LUKE Model

After fine-tuning a LUKE model with the curated training data described earlier on, we ran the model on the CoNLL-2003 test data set as well as the curated test set that we were working with. The results from these tests can be seen in Figure 1, Figure 6, and Figure 7.

Our fine-tuned LUKE model was able to identify names of different races at a very similar rate. The lowest-performing name group was Black/African-American names and the F1-PER score for this group was 0.9684. The difference

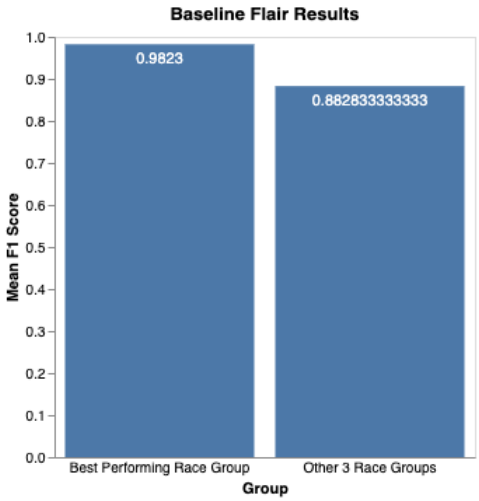

Figure 3: Baseline Flair Average Test Results

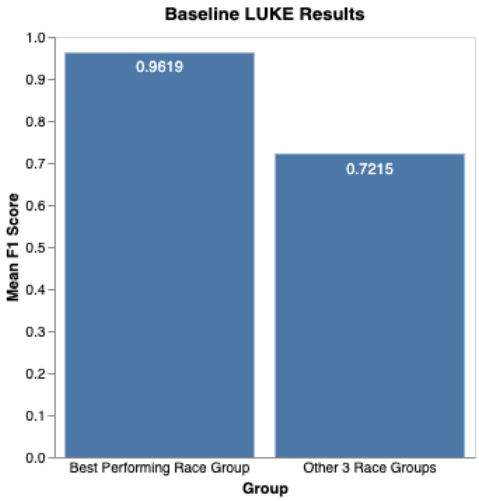

Figure 5: Baseline LUKE Average Test Results

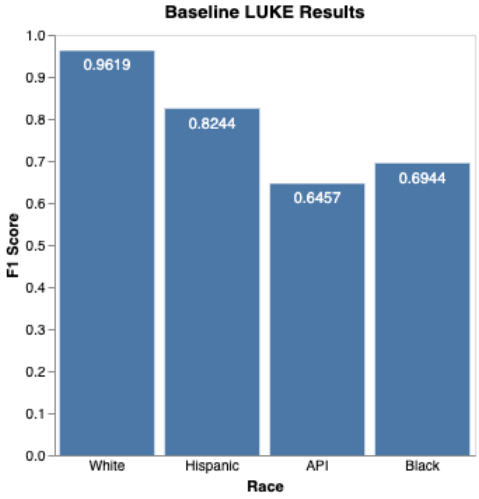

Figure 4: Baseline LUKE Test Results

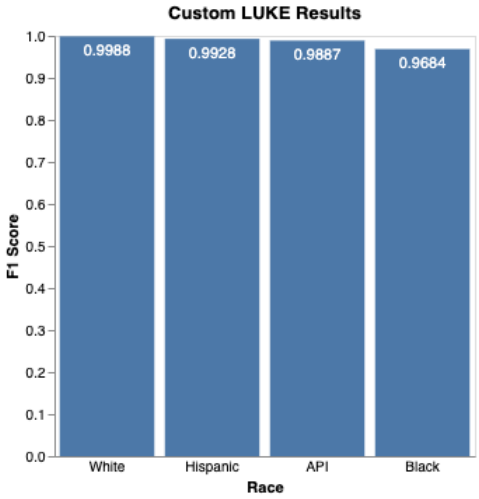

Figure 6: Custom LUKE Test Results

between that and the F1-PER score for the best performing group was only about 0.03. We were pleased to see that this difference between the best performing group's F1 score and worst performing group's F1 score was greatly reduced.

In Figure 1, we display the weighted average F1 score and the F1-PER score for the models on the CoNLL-2003 test dataset. The custom LUKE model achieves higher F1-PER and weighted average F1 scores than the baseline LUKE model. This shows that we did not have to sacrifice model performance to develop a model that performs more fairly across the race groups.

5 Conclusion

In this project, we analyzed the results of two state-of-the-art named entity recognition models: LUKE and Flair. We focused specifically on the NER models' ability to identify person entities and tested the models with a curated test set of names primarily belonging to four distinct racial groups. After noting the significant performance differences amongst the various name groups, we fine-tuned a LUKE model to mitigate the demonstrated bias. Our fine-tuned LUKE model outperformed the baseline LUKE model on the F1 weighted average score and the F1-PER score. Additionally, it greatly reduced the bias identified in the model.

Acknowledgments

We would like to thank the UC Berkeley MIDS W266 faculty for providing us with guidance throughout this project. We would also like to thank Peter Grabowski for reviewing this paper

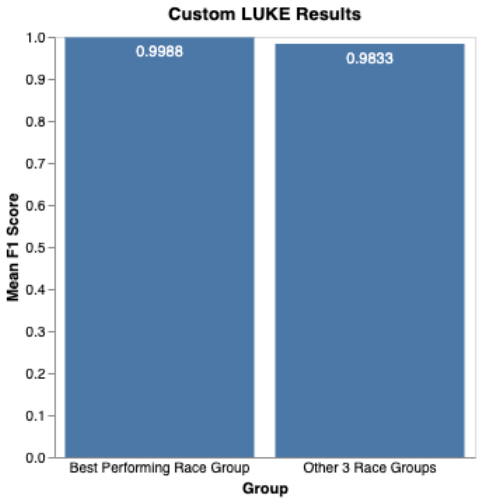

Figure 7: Custom LUKE Average Test Results

and assisting us throughout the project.

References

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, jun 2019. Association for Computational Linguistics.

[2] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, 2019.

[3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, aug 2018. Association for Computational Linguistics.

[4] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. jul 2016.

[5] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, 2019.

[6] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of Opportunity in Supervised Learning. oct 2016.

[7] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. jul 2019.

[9] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. Named Entity Recognition: Fallacies, challenges and opportunities. *Computer Standards and Interfaces*, 35(5), 2013.

[10] Shubhanshu Mishra, Sijun He, and Luca Belli. Assessing Demographic Bias in Named Entity Recognition. aug 2020.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014.

[12] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep

contextualized word representations. In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, 2018.

[13] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task. 2003.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. jun 2017.

[15] Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. CrossWeigh: Training named entity tagger from imperfect annotations. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2020.

[16] Vikas Yadav and Steven Bethard. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. oct 2019.

[17] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. 2020.