Instate: Predicting the State of Residence From Last Name *

Atul Dhingra† Gaurav Sood‡

January 30, 2023

Abstract

India has twenty-two official languages. To serve such a diverse language base is a challenge for survey statisticians, call center operators, software developers, and other such service providers. To provide better services and greater access to different language communities, we provide a way to use machine learning to improve localization by inferring the languages that are known to the user. To infer the language of the user, we rely on names. Using nearly 438M records spanning 33 states and 1.13M unique last names from the Indian Electoral Rolls Corpus (Sood and Dhingra, 2023), we build a character-level transformer-based machine-learning model that predicts the state of residence based on the last name. The model has a top-3 accuracy of 51.5% on unseen names. We further map the states to languages and provide a best guess for the languages understood by the respondent. We also provide open-source software that implements the method discussed in the paper.

Key words: localization, machine learning, rnn

1 Introduction

An overwhelming majority of the people in the world don't speak the language in which a vast majority of the web content is written and in which services are delivered. Hence, localization is essential for delivering software and services, especially to people in the third world. We tackle the problem of localization in a particularly challenging and populous country—India. India has twenty-two official languages. And serve such a diverse language base is a challenge for survey statisticians, call center operators, software developers, and other such service providers. To provide better services and greater access to different language communities, we provide a way to use machine learning to improve

*Data and scripts needed to replicate the results are available at: https://github.com/appeler/instate/

†Machine Learning Manager at Standard AI; dhingra.atul92@gmail.com

‡Independent researcher; gsood07@gmail.com.

localization by inferring the languages that are known to the user. To infer the language of the user, we rely on names. Using nearly 438M records spanning 33 states and 1.13M unique last names from the Indian Electoral Rolls Corpus (Sood and Dhingra, 2023), we build a character-level transformer-based machine-learning model that predicts the state of residence based on the last name. The model has a top-3 accuracy of 51.5% on unseen names. We further map the states to languages and provide a best guess for the languages understood by the respondent. We also provide open-source software that implements the method discussed in the paper.

2 Data

We exploit the Indian electoral rolls data (Sood and Dhingra, 2023, Sood et al., 2018) to build the machine learning model. The corpus has data from nearly 438 million people from 33 states with nearly 1.14 million unique last name spellings. The electoral roll data are at the polling station level. They include data on the elector, their husband or father's name, age, sex, and house number, along with details like the polling station, constituency, local police station, etc. Electoral rolls in India are a particularly useful source of data given they are a proximate census of the adult population. Unlike the US, in India, a neutral administrative body called the Election Commission works to produce an exhaustive list of eligible voters—all adult citizens. The aim to publish the electoral rolls online ahead of the elections is done with the purpose of making sure that no one is left out, and that others can point to errors of registering dead voters, etc. There are however weaknesses in the data. All too often, voters are registered in their 'home' districts, even if they have immigrated. (This works to our advantage to the extent that we rely on name, and state mappings to infer the languages that the user can understand.)

The parsed electoral rolls corpus (Sood et al., 2018) includes transliterations to English using the Python package indicate (Chintalapati and Sood, 2022). There is no unique English transliteration of Hindi names and the package provides the most probable transliteration.

We start by collating the data, preserving only three columns: first name, last name, and state. We assume the last word to be the last name if the last word is more than 2 letters. We remove all the data points where the Last name cannot be established directly or indirectly(by using Father or Husband's last name). We also remove all the names where special characters are inserted and aren't alphanumeric. We finally exclude names where the last name is present less than 3 times across the dataset given these last names are unlikely to be true last names. We finally convert all the last names into lowercase.

3 Models

3.1 Naive Bayes

We start by assuming that electoral rolls constitute the universe of people in India with the correct, unique transliterated English spellings of their names. Under those conditions, and if the last name is the only piece of information we have about the names, the Bayes Optimal classifier is simply an intercept-only model that gives the population mean—-the proportion of last names in various states.

3.2 KNN

Our second baseline is from the KNN model. We take the training set as our lookup corpus and for each test set example, we find the closest example in the training data. Our test set top-3 accuracy—chances that our top 3 predictions have the modal state from which people with that last name live—is 78%. The striking performance is a result of the fact that there are multiple transliterations of the same name and this means that we have for each test set entry, we have names with similar distribution in the training set.

3.3 RNN

The Naive Bayes model rests on untenable assumptions. It assumes no data entry errors, unique transliterations, and a universe of the adult population. The KNN model exploits 'data leakage.' To overcome these limitations we need a model that captures the patterns in names so that we can generalize out-of-sample. To that end, we build a character-level RNN model that takes the sequence of characters to predict the top three states with the highest conditional probability of people with the last name (Wu and He, 2020, Hua et al., 2018). In particular, we leverage a 2-layer RNN with 128 hidden units, train the model for 1M epochs using SGD optimizer, and use a negative log-likelihood loss. To deal with exploding gradients, we set a low learning rate of 0.005, and clip gradients with a hyper-parameter of .25.

The data is split on last names with an 80-20 split such that last names in training and test sets are unique so that we can understand the performance on unseen names. On the test set, the top-3 accuracy—-chances that our three predictions include the modal state in which people with that last name live—of our model is 51.5%. Compare this to the no-information baseline where we only know the population of the states and guess the modal state. The most populous state in India is Uttar Pradesh and if we were to guess that as the answer for all the names, we would be right less than 16.74% of the time.

4 States to Languages

While the modern Indian state boundaries and state education curricula are affected by language conflicts (for instance, see this article), there is no unique mapping between states and languages. This is for a few reasons. For one, many Indian states are massive and there is considerable linguistic diversity across regions within states. Second, massive internal migration, especially to urban centers, has transformed many cities. This compromises our data. For instance, say that all Dhingras only speak Punjabi and say that all of them once upon a time lived in Punjab. Say that over time half of them immigrated to Delhi and never learned another language. Say that we map Delhi to people who understand Hindi. Our classifier will then say that Dhingras have a 50% chance of speaking Punjabi and a 50% chance of speaking Hindi. But in fact, no Dhingra can understand Hindi. Our prediction in such a case would be highly erroneous. We have reasons to think, however, that the net error is generally likely lower. For instance, in the above example, we make a slightly untenable assumption that immigrants never learn to understand the dominant language of the geography they immigrate to. This can be true but it is also likely that there is both selection—Dhingras who are open to learning new languages or may already know another language are likelier to immigrate to Delhi—and learning—Dhingras who immigrate to Delhi learn a new language over time.

To create our states to languages database, we consulted the state constitutions for the official languages and the latest Indian census (the 2011 census) for the most widely spoken languages in the state.

5 Discussion

Our paper makes a contribution to improving access to software via localization. We provide a Python package that takes the last name of a person and gives its distribution across states.

Our method of predicting language from the last name has a few limitations. The first is that because of massive internal migration, rising standardization of the language of education, etc., the relationship between geography and languages that are understood by the person living in the geographic area is increasingly weak. The second issue is that our geographical units are very crude. Indian states are often as big as countries. For instance, Uttar Pradesh has more than 230 million people. There is also large linguistic diversity within states with people in urban areas, for instance, more likely to understand English, and more likely to have immigrants from other areas of India. In future versions of the software, we hope to exploit the more granular data that is provided in the electoral rolls. The third challenge is how to incorporate transliterations. There is no one standard way of transliterating Indian languages to English. We use one transliteration when the

way to improve coverage may be to use k-best transliterations with the transliterations produced roughly in the frequency of how common they are. The fourth challenge is in establishing last names. Our method of establishing the last name is crude. One way to improve the heuristic is to only use last words in names that are common across the household members' names.

References

CHINTALAPATI, R. AND G. SOOD (2022): "Indicate: Transliterate Indic Languages to English," .

HUA, Q., S. QUNDONG, J. DINGCHAO, G. LEI, Z. YANPENG, AND L. PENGKANG (2018): "A Character-Level Method for Text Classification," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, 402–406.

SOOD, G. AND A. DHINGRA (2023): "Indian Electoral Roll PDF Corpus," .

SOOD, G., S. LAOHAPRAPANON, M. SANJEEVI, AND K. TRAN (2018): "Parsed Indian Electoral Rolls," .

WU, X. AND J. HE (2020): "Character-Level Recurrent Neural Network for Text Classification Applied to Large Scale Chinese News Corpus," in *2020 The 3rd International Conference on Machine Learning and Machine Intelligence*, New York, NY, USA: Association for Computing Machinery, MLMI '20, 83–87.

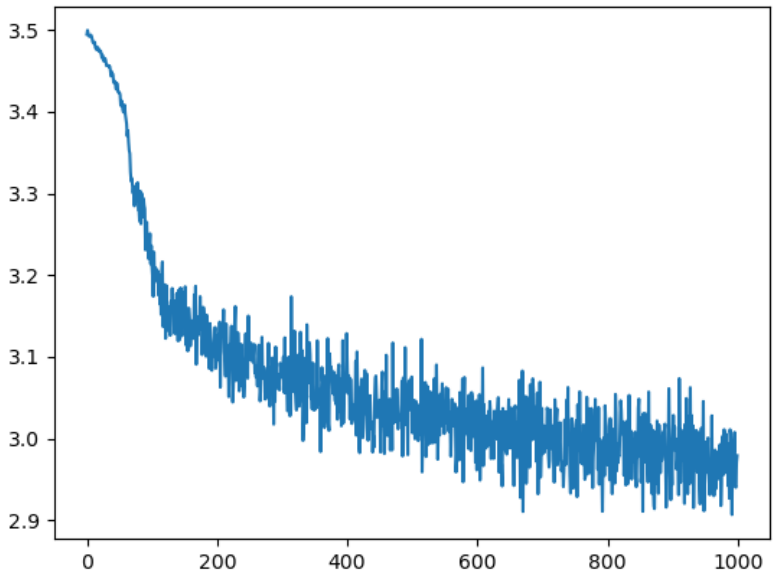

Figure 1: Negative Log Loss on the test set.