Extracting Information from Indian First Names

Akshay Gulati
Dept. of Electronics&Comm.
Jamia Millia Islamia
New Delhi – 110025, India
akshaygulati@robot-maker.net

Abstract

First name of a person can tell important demographic and cultural information about that person. This paper proposes statistical models for extracting vital information that is gender, religion and name validity from Indian first names. Statistical models combine some classical features like n-grams and Levenshtein distance along with some self observed features like vowel score and religion belief. Rigorous evaluation of models has been performed through several machine learning algorithms to compare the accuracy, F-Measure, Kappa Static and RMS error. Experimental results give promising and favorable results which indicate that these models proposed can be directly used in other information extraction systems.

1 Introduction

Name validity, gender and religion of a person can be successfully predicted to a certain extent just by his first name. But, as there is no direct relationship between the first names and these entities an absolute prediction is impossible. Therefore, the proposed statistical models aim to develop an indirect relationship between the first names and these entities for extracting information from first names.

Theoretically, a first name is a proper noun, which means that a first name can be any sequence of alphabets. Therefore, absolute name validity is impossible as any sequence of alphabets is a valid first name. For practical purposes, this paper assumes that most of the Indian first names are the ones which have some historical/cultural/ethnic relevance and are not some arbitrary sequence of alphabets. Upon assuming this, statistical models proposed are constructed and are used with machine learning algorithms for training classifiers. The trained classifier can then be used for predicting the validity of a first name that is differentiating between a valid first name and an invalid first name.

Even absolute gender and religion predictions are impossible as there is no restriction on the naming process with regards to gender and religion. A person of any gender and any faith can identify himself/herself with any name of his/her choice. For example, a Hindu girl can name herself John without breaking any law. For practical purposes, such cases have been left out for construction of the statistical models proposed.

The models proposed along with the machine learning algorithms can find direct use in information extraction systems and real time applications such as:

i. Automatic field suggestions for form filling.
ii. Anomaly detection in pre-filled forms.
iii. Analyzing demographic and religious trends/sentiments from social media collected data based on first names.
iv. Filtering forms/application with respect to certain gender or religion.
v. Filtering spam/fake accounts on internet by name validity.

Other demographic information can also be extracted from Indian first name such as expected age group, caste of person and part of India (north/south/east/west). These topics have been left for future research purposes.

2 Data

The training and testing data used has been collected from public Indian name databases available on the internet.

D S Sharma, R Sangal and E Sherly. Proc. of the 12th Intl. Conference on Natural Language Processing, pages 138–143, Trivandrum, India. December 2015.

For name validity, the first names have been classified into 'Valid' and 'Invalid' classes. A total of 8970 first names are used in the training data out of which 7846 are valid first names. 650 are noisy words and 475 are completely random sequence of alphabets, together making invalid names.

For gender prediction, the first names have been classified into 'Male' and 'Female' classes. A total of 7846 first names are available in the training data out of which 4509 are male first names and 3337 are female.

For Religion prediction, the first names have been classified into 'Hinduism', 'Islamic' and 'Christian' classes. A total 7846 first names used in the training data out of which 3758 are 'Hinduism' first names, 3501 are 'Islamic' first names and 587 are 'Catholic' first names. The numbers are in proportion of diversity of first names of each religion in India. 'Sikhism' as class for prediction was not considered as many of its first names are common with 'Hinduism' and this causes ambiguity.

The dataset is available for download[1] and can be used for future research regrading Indian names with appropriate citations.

3 Content Models

Before applying machine learning algorithms it is important to convert the training data available into content models. These content models contain different features which have been developed or chosen after careful observation and evaluation of the training data.

3.1 Name Validity Models

This paper proposes four different name validity models for experiments. Each model contains two features for each first name in the training data. First feature is vowel score and is model independent. The second feature is the n-gram (Manning and Schütze, 1999) score for n = 1 or 2 or 3 or skipping bigrams, each for a different model. An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a $(n-1)$–order Markov model (Baum and Petrie, 1966).

Vowel score tells us the average vowel distance between the vowels in a first name and is normalized with respect to the length of first name. It is an important feature based on the observation that for any first name, occurrence of vowels will be similar to all the other first names in the training data. As first names are sequences of letter with low vowel count averaging about 2-3 vowels per first name, the vowel score gives us a good idea of the vowel patterns in the first names in general. Therefore, any anomaly in the vowel occurrence will directly lead us to the result that the given first name is invalid. On the contrary, if there is no anomaly in vowel occurrence then other features like n-grams will be used.

Mathematically, the vowel score for a 'Name' is given by *VS(Name)*, where '*n'* is the length of the given first name and *Distance(i)* is the distance of the i^{th} vowel from the $(i\text{-}1)^{th}$ vowel in the first name.

$$VS(Name) = \frac{\sum_{i=2}^{i=n}[Distance(i)]}{(No.of\ vowels)\ X\ (\ |Name|)}$$

Unigram is a n-gram where the size of token is 1. Unigrams help provide the probability of a token; it assumes the position of token to be independent of other tokens in the first name.

Mathematically, unigram score for a 'Name' is given by *U(Name)*, where '$Name_i$' is the i^{th} token/letter of the given first name, '*n'* is the length of the given first name, *Token* is the set of the 26 alphabets and '$Token_j$' is the j^{th} token/letter of the set *Token*.

$$U(Name) = \prod_{i=1}^{i=n} \frac{Count(Name_i)}{\sum_{j=1}^{j=26} Token_j}$$

Bigram is a n-gram where the size of token is 2. Bigrams help provide the conditional probability of a token given the preceding token has occurred.

Mathematically, bigram score for a 'Name' is given by *B(Name)*, where '$Name_i$' is the i^{th} token/letter of the first name given that one start symbols (*) was added before each first name and an end symbol (#) was added after each first name before experiment, '*n'* is the length of given first name.

$$B(Name) = \prod_{i=2}^{i=n} \frac{Count(Name_{i-1}, Name_i)}{Count(Name_{i-1})}$$

Trigram is a n-gram where the size of token is 3. Trigrams help provide the conditional

[1] http://www.robot-maker.net/wp-content/uploads/2015/10/Indian_Unified_Names.txt

probability of a token given the preceding token has occurred.

Mathematically, trigram score for a 'Name' is given by *T(Name)*, where '$Name_i$' is the i^{th} token/letter of the first name given that two start symbols (*) were added before each first name and an end symbol (#) was added after each first name before experiment, 'n' is the length of the given first name.

$$T(Name) = \prod_{i=3}^{i=n} \frac{Count(Name_{i-2}, Name_{i-1}, Name_i)}{Count(Name_{i-2}, Name_{i-1})}$$

Skipping Bigram (Xuedong *et al*, 1992) is a special n-gram where size of token is 2 and a condition that first character of a bigram can't be a vowel when the second character of bigram is a consonant. Skipping bigrams were used on careful observations of training data, it allows us to use only relevant bigrams and ignore the redundant ones.

Mathematically, the skipping bigram score is calculated by the same formula as for bigrams if only valid tokens are considered and invalid tokens are skipped.

3.2 Gender Prediction Models

This paper proposes four models for gender prediction. Each model contains two features for each first name in the training data, these features keep count of some specific tokens (listed below) which occur at the name endings for male and female first names respectively. The tokens are explained below, each for a different model.

Unigram model to keep count of all the unigrams (in training data) such that the only character of unigrams is the last character of the first name.

Bigram model to keep count of all the bigrams (in training data) such that the 2 characters of bigrams are the last two characters of the first name.

Trigram model to keep count of all the trigrams (in training data) such that the 3 characters of trigrams are the last 3 characters of the first name.

Vowel Bigram model to keep count of all the bigrams (in training data) such that the first character of bigram is the last vowel and the second character of bigrams is the last character of the first name.

3.3 Religion Prediction Model

This paper proposes one model for the religion prediction. The idea behind the design of religion prediction model is to calculate the similarity/closeness of a given first name with other first names of each religion in the training data. The concept of Levenshtein distance (Vladimir, 1966) is used to calculate the minimum edit distance between two first names. This model contains three features for each first name in the training data, these features are the Hinduism Belief, Islam Belief and Christianity Belief. These three features can be extracted from a single function *Belief(r)*.

Mathematically, belief for religion 'r' is given by *belief(r)*, where *Count(r,i)* gives the total no. of first names of religion '*r*' in training data with levenshtein distance equal to '*i*' for the first name for which the features are being calculated and '*Depth*' is an experimentally chosen quantity whose value is to be taken equal to 3. *'Depth'* parameter sets maximum value of levenshtein distance for which *Belief(r)* will be calculated.

$$Belief(r) = \sum_{i=1}^{i=Depth} \frac{Count(r,i)}{i^2}$$

Mathematically, the Levenshtein distance between two strings 'a' and 'b' is given by $lev_{a,b}(|a|,|b|)$, where $1_{(a_i \neq b_i)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1 otherwise.

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & if\ \min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_i)} \end{cases} & otherwise \end{cases}$$

4 Evaluation

The proposed statistical models are constructed in Python using the training data. Further, Weka (Hall *et al*, 2009) was used to apply various machine learning algorithms on the content models for training classifiers. Weka is an open source collection of machine learning algorithms. For all algorithms, 10 fold cross validation was done. Detailed results comprising of accuracy, F-Measure (Powers, 2011), Kappa Static and RMS error have been represented below in tabular form.

	LAD Tree	REP Tree	Random Forest	**Bagging**	Simple Cart
% Correct	96.3371	96.427	96.4607	**96.6854**	96.5843
% Incorrect	3.6629	3.573	3.5393	**3.3146**	3.4157
Precision	0.963	0.963	0.964	**0.966**	0.965
Recall	0.963	0.964	0.965	**0.967**	0.966
F-Measure	0.963	0.963	0.964	**0.966**	0.965
Kappa Static	0.8215	0.8223	0.8275	**0.8361**	0.8294
RMS Error	0.1639	0.1672	0.1637	**0.1596**	0.169

Table 1. Results of Unigram model for name validity

	LAD Tree	**REP Tree**	Random Forest	Bagging	Simple Cart
% Correct	98.1278	**98.1614**	97.7803	98.1502	98.0493
% Incorrect	1.8722	**1.8386**	2.2197	1.8498	1.9507
Precision	0.981	**0.982**	0.978	0.981	0.98
Recall	0.981	**0.982**	0.978	0.982	0.98
F-Measure	0.981	**0.982**	0.978	0.981	0.98
Kappa Static	0.9104	**0.9136**	0.8953	0.9125	0.9071
RMS Error	0.1224	**0.0286**	0.13	0.1196	0.0304

Table 2. Results of Bigram model for name validity

	LAD Tree	**REP Tree**	Random Forest	Bagging	Simple Cart
% Correct	96.1371	**96.4282**	95.7452	96.3722	96.4058
% Incorrect	3.8629	**3.5718**	4.2548	3.6278	3.5942
Precision	0.96	**0.963**	0.957	0.963	0.963
Recall	0.961	**0.964**	0.957	0.964	0.964
F-Measure	0.961	**0.964**	0.957	0.963	0.963
Kappa Static	0.8135	**0.8277**	0.7988	0.8252	0.8259
RMS Error	0.1693	**0.1703**	0.1795	0.166	0.1723

Table 3. Results of Trigram model for name validity

	LAD Tree	**REP Tree**	Random Forest	Bagging	Simple Cart
% Correct	97.8885	**97.9667**	97.5869	97.855	97.8773
% Incorrect	2.1115	**2.0333**	2.4131	2.145	2.1227
Precision	0.979	**0.979**	0.976	0.978	0.979
Recall	0.979	**0.98**	0.976	0.979	0.979
F-Measure	0.979	**0.98**	0.976	0.978	0.979
Kappa Static	0.9022	**0.9055**	0.8874	0.9	0.9016
RMS Error	0.1278	**0.1293**	0.1373	0.1278	0.1393

Table 4. Results of Skipping Bigram model for name validity

	LAD Tree	REP Tree	**Regression**	Bagging	Simple Cart
% Correct	75.2176	75.5163	**75.5683**	75.5553	75.5683
% Incorrect	24.7824	24.4837	**24.4317**	24.4447	24.4317
Precision	0.762	0.768	**0.768**	0.768	0.768
Recall	0.752	0.755	**0.756**	0.756	0.756
F-Measure	0.754	0.757	**0.757**	0.757	0.757
Kappa Static	0.5012	0.5089	**0.5097**	0.5095	0.5097
RMS Error	0.4095	0.4221	**0.4089**	0.4174	0.4222

Table 5. Results of Unigram model for gender prediction

	LAD Tree	REP Tree	**Regression**	Bagging	Simple Cart
% Correct	82.4823	82.7395	**83.0482**	82.7524	82.881
% Incorrect	17.5177	17.2605	**16.9518**	17.2476	17.119
Precision	0.826	0.829	**0.832**	0.829	0.83
Recall	0.825	0.827	**0.83**	0.828	0.829
F-Measure	0.825	0.828	**0.831**	0.828	0.829
Kappa Static	0.642	0.6476	**0.6546**	0.6477	0.6504
RMS Error	0.3625	0.3703	**0.358**	0.3668	0.3623

Table 6. Results of Bigram model for gender prediction

	LAD Tree	REP Tree	Regression	Bagging	Simple Cart
% Correct	**86.8424**	86.4201	86.8168	86.1001	86.1897
% Incorrect	**13.1576**	13.5799	13.1832	13.8999	13.8103
Precision	**0.868**	0.864	0.868	0.861	0.862
Recall	**0.868**	0.864	0.868	0.861	0.862
F-Measure	**0.868**	0.864	0.868	0.861	0.862
Kappa Static	**0.7303**	0.7221	0.73	0.7158	0.717
RMS Error	**0.3128**	0.3188	0.3072	0.3138	0.3197

Table 7. Results of Trigram model for gender prediction

	LAD Tree	REP Tree	Regression	**Bagging**	Simple Cart
% Correct	80.5624	81.7361	80.7558	**82.1488**	81.1815
% Incorrect	19.4376	18.2639	19.2442	**17.8512**	18.8185
Precision	0.813	0.824	0.814	**0.828**	0.817
Recall	0.806	0.817	0.808	**0.821**	0.812
F-Measure	0.807	0.818	0.809	**0.823**	0.813
Kappa Static	0.6081	0.6312	0.6114	**0.6395**	0.6193
RMS Error	0.3815	0.3658	0.3674	**0.3541**	0.3678

Table 8. Results of Vowel Bigram model for gender prediction

	J48	REP Tree	**Regression**	Bagging	Simple Cart
% Correct	82.2321	82.2704	**83.2908**	81.9388	82.1556
% Incorrect	17.7679	17.7296	**16.7092**	18.0612	17.8444
Precision	0.818	0.817	**0.829**	0.814	0.817
Recall	0.822	0.823	**0.833**	0.819	0.822
F-Measure	0.816	0.817	**0.826**	0.814	0.815
Kappa Static	0.6785	0.6795	**0.6963**	0.6739	0.6764
RMS Error	0.3044	0.3001	**0.289**	0.2937	0.3032

Table 9. Results of Religion Prediction model

5 Conclusion

The models proposed for different topics have performed well overall. The results have been tabulated and shown for comparison of the different types of models proposed.

i. From the results obtained, it can be seen that Bigram model has performed the best with accuracy of 98.1614% closely followed by the Skipping Bigram model, both using REP Tree algorithm.

ii. Due to sparse data the Trigram model has not performed well and its results are comparable to the Unigram model.

iii. For gender prediction, Trigram model has the best with the accuracy of 86.8424% with LAD Tree algorithm.

iv. Trigram model has outperformed other models not only because it is more detailed but also it has better chances for handling exception cases.

v. For religion prediction, the *belief(r)* has been proved to be an effective feature for calculating belief as the model has performed with an accuracy of 83.44%.using Regression algorithm.

In conclusion, the proposed models have been successfully tested through rigorous evaluations and have completed the objectives of the research. They can further be used in other research areas and real time applications as mention in the introduction.

Reference

L.E Baum and T. Petrie. 1966. *Statistical Inference for Probabilistic Functions of Finite State Markov Chains*. The Annals of Mathematical Statistics, Vol 37 (6): 1554–1563.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press: ISBN 0-262-13360-1 : 191-199.

Xuedong Huang, Fileno Alleva, Hsiao-wuen Hon, Mei-yuh Hwang and Ronald Rosenfeld. 1992. *The SPHINX-II Speech Recognition System: An Overview*. Computer, Speech and Language, Vol 7(1): 137-148.

Vladimir I. Levenshtein. 1966. *Binary codes capable of correcting deletions, insertions, and reversals*. Soviet Physics Doklad, Vol 10 (8): 707–710.

David M W Powers. 2011. *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation*. Journal of Machine Learning Technologies, Vol 2 (1): 37–63.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. 2009. *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, Vol 11 (1).