Malaysian Name-based Ethnicity Classification using LSTM

Youngbum Hur[1*]
[1] Department of Industrial Engineering, Inha University
Incheon, 22212 South Korea
[e-mail: youngbum.hur@inha.ac.kr]
[*]Corresponding authors: Youngbum Hur

Received June 7, 2022; revised October 19, 2022; accepted December 8, 2022; published December 31, 2022

Abstract

Name separation (splitting full names into surnames and given names) is not a tedious task in a multiethnic country because the procedure for splitting surnames and given names is ethnicity-specific. Malaysia has multiple main ethnic groups; therefore, separating Malaysian full names into surnames and given names proves a challenge. In this study, we develop a two-phase framework for Malaysian name separation using deep learning. In the initial phase, we predict the ethnicity of full names. We propose a recurrent neural network with long short-term memory network–based model with character embeddings for prediction. Based on the predicted ethnicity, we use a rule-based algorithm for splitting full names into surnames and given names in the second phase. We evaluate the performance of the proposed model against various machine learning models and demonstrate that it outperforms them by an average of 9%. Moreover, transfer learning and fine-tuning of the proposed model with an additional dataset results in an improvement of up to 7% on average.

Keywords: Deep Learning, Recurrent Neural Network, LSTM, Machine Learning, Ethnicity Classification, Malaysian Name Separation, Deep Learning-based Name Separation.

http://doi.org/10.3837/tiis.2022.12.004 **ISSN : 1976-7277**

1. Introduction

Names are important demographic categorization of people. In most regions and time periods, surnames were based on descent from a male ancestor; therefore, names can represent history and diversity of culture.

In a homogeneous society, all individuals share the same racial ethnicity, language, and a series of beliefs (*e.g.*, South Korea); however, not all countries are composed of homogeneous societies. For example, Malaysia has multiple main ethnic groups. There are various genetic, linguistic, cultural, and social categories among the Malaysian subgroups because of several years of immigration and assimilation of people with multiple regional ethnicities. The ethnicity distribution of Malaysia is shown in **Fig. 1**; 51% are Malay/Muslim, 24.2% are Chinese, and 7.2% are Indian [24].

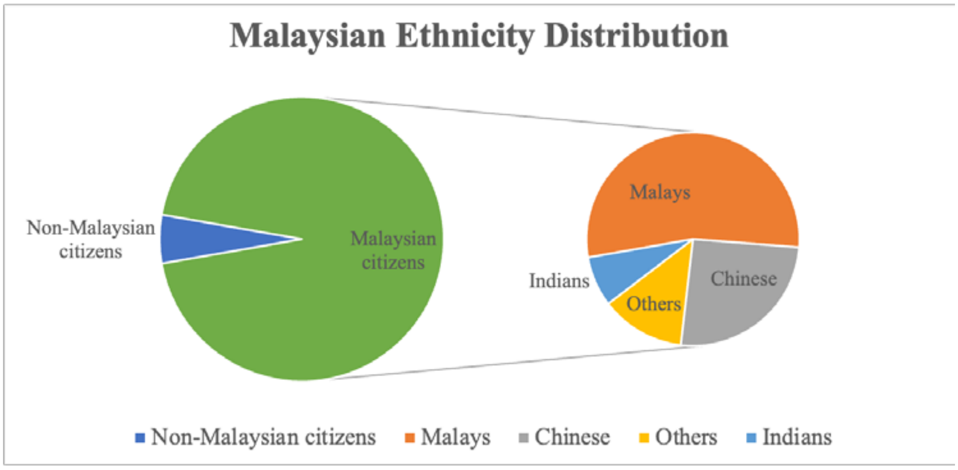


Fig. 1. Ethnicity distribution of Malaysia

Malaysian passports have full names, but surname and given name fields are not differentiated on the passport because they do not follow the International Civil Aviation Organization (ICAO) standards which require the name after the three letters country code should be the surname of the passport holder. This causes difficulties or confusion in some countries. For instance, the second example in **Fig. 2** shows that surname "SMITH" follows the country code "AUS", which stands for Australia. The symbol << separates the surname from given names. Therefore, we can easily identify this person's full name as "John William Smith," in which "Smith" is the surname. However, as shown in the first example in **Fig. 2**, Malaysian passport does not have the delimiter <<. Thus, given names are often not identified by a machine automatically.

P<MYSALI<AKBAR<BIN<MOHAMAD<

P<<AUSSMITH<<JOHN<<WILLIAM<

Fig. 2. Examples of names on Malaysian and Australian passports

This motivates us to separate full names into surnames and given names. We propose a two-phase framework for name separation instead of directly splitting full names into surnames and given names because of the following two reasons. First, there is a lack of labeled data that separates surnames and given names. Second, there are specific name separation rules for each ethnicity. For instance, for Malay/Muslim ethnicity, if a name contains the keyword “BINTE”, we consider words before and after the keyword as first and last names, respectively. We must know the ethnicity for each full name to apply these name-separation rules. Once we know the ethnicity of a name is determined, there is no uncertainty in the separation of surnames and given names because predefined name-separation rules are rule-based. Therefore, this problem is similar to the ethnicity-classification problem. Section 3 provides a detailed description of this problem.

In this study, we construct a labeled training dataset (full names with corresponding ethnicities) from scratch and employ a two-phase framework to address this problem. We propose an RNN-LSTM model with character embeddings for identifying ethnicity to separate given names and surnames. The main contributions of this study are as follows.

- We propose a two-phase framework for separating Malaysian names into surnames and given names.
- We develop an RNN-LSTM model with character embeddings for predicting ethnicities.
- We construct labeled ethnicity data from scratch using two approaches.
- We conduct experiments using various machine learning models for ethnicity classification.
- We use transfer learning and fine-tuning using an additional dataset for further improvement of the model.

The remainder of this paper is organized as follows. Section 2 reviews relevant existing work. Section 3 describes the problem in detail. Section 4 explains the machine learning models we use. Section 5 presents the experimental details and discusses the results. We discuss some future studies, which can enhance the proposed model by utilizing unlabeled data in Section 6. Finally, Section 7 presents the concluding remarks.

2. Literature Review

2.1 Ethnicity classification and name separation

Ethnicity identification through names has been studied using several applications, and perhaps biomedical research is an area that has studied this problem the most [1]. The genetics of dietary differences on each race or ethnicity is significant. Buechley [2] utilized the Generally Useful Ethnicity Search System (GUESS) to determine Hispanic ethnicity corresponding to Spanish names. Coldman et al. [3] classified names as either being Chinese or non-Chinese. Some researchers have considered Western names for ethnicity classification and used a Bayesian approach for prediction [4].

Dictionary-based surname methods have been widely used for ethnicity classification. However, these methods fail when they encounter names that are not in the dictionary. To address this problem, machine learning algorithms have been used for ethnicity classification. Ambekar et al. [5] combined decision tree and hidden Markov models (HMMs) to conduct classification on a taxonomy with 13 leaf classes. Treeratpituk and Giles [6] utilized both alphabet and phonetic sequences in names to improve performance and applied it to analyze the evolution of ethnicities in the computer science research community.

The most similar research to our paper is [26]. Wong et al. [26] proposed a framework to predict ethnicity using personal name and census location in Canada. They used machine learning algorithms such as regularized logistic regression and C-SVM.

The work on name separation using machine learning is much more limited than the work for ethnicity classification using machine learning. Kim et al. [25] tackled the problem of distinguishing authors who have the same names (*i.e.*, same name means same first forename initial and full surname) in bibliographic. They used ethnicity information for better disambiguation performance.

Unlike computer vision community, there are no datasets that are commonly used in name-based ethnicity classification researches. Therefore, many researchers often use their own specific datasets, so it makes us hard to find a similar work. To the best of our knowledge, our research is the first to predict Malaysian ethnicity using machine learning algorithms.

2.2 Recurrent neural networks and long short-term memory

Text classification is a subfiled of natural language processing (NLP) tasks with real-world applications such as spam detection and fraud transaction detection [7–9]. Recurrent neural networks (RNN) are known for effectively predicting sequential data, such as natural language. Mikolov et al. [10] demonstrated that RNN is usefule for a language modeling. Bahdanau et al. [11] used RNN for machine translation, which showed much better performance than the statistical machine translation models. However, RNN is adversely affected by long-term dependency because of its recurrent structure [12] as well as overfitting problems [13].

To address this issues, long-term dependencies, long short-term memory (LSTM) was proposed to reduce the long-term dependency problem using a memory cell and forget gate [14]. Similar RNN cells such as gated recurrent unit (GRU) [15] were introduced to improve the efficiency of LSTM. Overfitting problems of RNNs were handled by applying dropout on non-recurrent connections of RNNs [13]. Therefore, RNN performed the best on sequential data, specifically in natural language processing tasks such as text classification.

Several researchers have focused on identifying nationalities and ethnicities. Lee et al. [16] proposed a recurrent neural network (RNN) model to predict nationalities of each name using automatic feature extraction. They utilized Olympic Games records for collecting names and considered 12 ethnicities. Their algorithm outperformed random forest and logistic regression. Yao et al. [17] utilized bidirectional RNN with LSTM for Chinese word segmentation. They tested their model with simplified Chinese and traditional Chinese dataset. However, studies on ethnicity prediction based on Malaysian names remain limited. To the best of our knowledge, this study represents the first attempt at Malaysian ethnicity classification and name separation using a deep learning method.

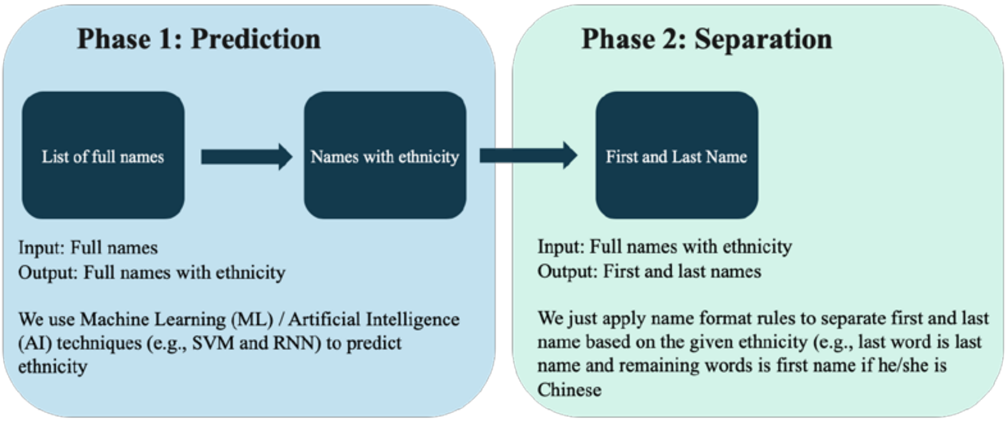
2.3 Transfer learning and fine-tuning for text classification

Transfer learning or domain adaptation is crucial in various natural language processing applications. Transfer learning is a machine learning technique where a pretrained model is reused as the starting point for a model on a new task. Language model pretraining is effective for improving several natural language processing tasks.

Dai and Le [18] used unlabeled data to improve sequence learning with recurrent networks and showed that the parameters got from the unsupervised step could be used as a starting point for other supervised training models for NLP tasks. Howard and Ruder [19] proposed an effective transfer learning method that can be applied to any task in NLP and introduced techniques to keep previous knowledge and avoid catastrophic forgetting during fine-tuning of a language model. Mou et al. [20] found that although similar pretraining tasks transfer better for NLP task, fine-tuning fails for unrelated tasks. Nevertheless, fine-tuning has been successfully used to transfer between similar tasks, such as question answering [21]. Additionally, Dai and Le [18] fine-tuned a language model but the mode was quite overfitted with 10,000 labeled examples and required millions of in-domain documents for good performance.

3. Problem Description

Fig. 3 shows a high-level scheme of the proposed two-phase framework to separate full names into surnames and given names. First, we identify the ethnicity corresponding to the full names. Subsequently, we separate surnames and given names using a rule-based algorithm based on the given ethnicity.



Input: Full names
Output: Full names with ethnicity

We use Machine Learning (ML) / Artificial Intelligence (AI) techniques (e.g., SVM and RNN) to predict ethnicity

Input: Full names with ethnicity
Output: First and last names

We just apply name format rules to separate first and last name based on the given ethnicity (e.g., last word is last name and remaining words is first name if he/she is Chinese

Fig. 3. High-level scheme of the proposed two-phase framework for name separation

In this study, we consider five ethnicities, “Malay/Muslim”, “Indian”, “Chinese”, “Western”, and “Other”. However, we do not have prior information on given names and surnames or the corresponding ethnicities. Therefore, we do not have labeled data, and this represents our primary challenge.

Sufficient size of training data is essential for obtaining a well-trained deep learning model. Therefore, we design two approaches for automatic data labeling. The first approach uses unique keywords for specific ethnicities. For instance, if a name contains a unique keyword relating to a specific ethnicity, we label it with the corresponding ethnicity. The following are some keywords that were used in this study.

- Indian: S/O (son of), D/O (daughter of), A/L, AL (anak lelaki), A/P, AP (anak perempuan), A/K, AK or W/O.
- Malay/Muslim: BIN, BTE, BINTE, BT, B., YB, Y.BHG., Y.A.M., Y.M., TAN SRI, SRI PADUKA, TUNKU, MAJOR, GENERAL, PUAN SRI, DATO, DATIN, DATUK, HAJI, HJ, HAJJAH, HJH, PENGIRAN, PINGGIRAN, UNGKU, NEE
- Others: U, KO, MAUNG, DAW, MA

However, only 7% of the data are labeled after applying the unique keyword approach. Therefore, we utilize an additional approach to automatically obtain more labeled data. The second approach uses common surnames. First, we collect a list of common surnames from publicly available sources [5] and count the ethnicities associated with the full names using a list of surnames for all ethnicities. The number of associated ethnicity can be more than one (*e.g.*, Ma could be a Chinese or Vietnamese surname). If there is only one associated ethnicity, we label that name with the corresponding ethnicity. If there are multiple associated ethnicities, we do not annotate it and leave it as unlabeled data. 38% of data, which is approximately 23,000 full names with ethnicities are labeled by applying the common surname approach. Thus, we use these names with known ethnicity for training the proposed model.

Training dataset 2 and 3 are the results of applying the first and second approaches, respectively, as listed in **Table 3**. It is worth noting that some level of label noise exists for the collected training dataset despite the presence of some labeled data. We believe we must have a clean test dataset for precisely evaluating the proposed method. Therefore, we created a test dataset with the corresponding ethnicities that were manually annotated by native speakers instead of using auto-labeling approaches.

4. Model Development

4.1 Various machine learning models

First, we consider various machine learning models to select the best models for comparison with the proposed model. A brief description of each model is as follows.

We consider (a) support vector machine: linear classifiers that are based on the margin maximization principle. In addition, we consider (b) boosting: an iterative strategy for adjusting the weight of an observation based on the previous classification; (c) bagging: a method in order to reduce prediction variance by producing additional data for training from dataset by combining repetitions with combinations to create multiset of the original data; (d) random forest: a classification algorithm using several decisions trees. It uses bagging and feature randomness when constructing each tree to create an uncorrelated forest of trees whose prediction by voting is more accurate than an individual tree; (e) neural network: a single hidden layer neural network; (f) decision tree: classification or regression tree. The parameters used in these models are listed in **Table 1** and the results are listed in **Table 2**.

Table 1. Parameters for machine learning algorithms

Associated model	Parameter name	Value
SVM	Cost	100
SVM	Kernel	radial
Neural Nets	Number of units	1
Neural Nets	Weight decay	0.0005
Random Forest	Number of trees	200

Table 2. Comparison of machine learning algorithms

Model	Accuracy
Support Vector Machine (SVM)	0.75
Boosting	0.733
Bagging	0.673
Random Forest	0.73
Neural Nets	0.736
Decision Tree	0.596

4.2 Support vector machine

We select support vector machine (SVM) as our baseline because it exhibits the best performance. SVM determines separators in the search space that can best separate the different classes. We have multiple classes to classify; therefore, we have one-against-one approach for multiclass classification with k(k-1)/2 binary classifiers. Notably, k is the number of classes. The appropriate class is selected using a voting scheme, and the soft-margin SVM formulation for binary classification is as follows.

$$\text{minimize} \quad ||w||^2 + C \cdot \textstyle\sum_i^N \xi_i \tag{1}$$

$$\text{subject to} \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad \text{for } 1 = 1, \ldots, N \tag{2}$$

In the second term of objective function, the slack variable ξ_i makes the input x_i to be closer to the hyperplane, but it is unavoidable to have some penalties in the objective function. If C is large enough, the SVM becomes considerably strict and attempts to obtain all points to be on the right side of the hyperplane. If C is small enough, the SVM becomes considerably less strict and may ”sacrifice” some points to obtain a simpler (*i.e.*, lower $||w||^2$) solution.

4.3 Recurrent neural network with long short-term memory network (RNN-LSTM)

Recurrent neural network (RNN) is a widely used deep learning model for text classification. In a basic RNN, the hidden states of the neural networks are updated as follows.

$$h_t = \begin{cases} 0 & t = 0 \\ f(h_{t-1}, x_t) & \text{otherwise} \end{cases}$$

Especially, x_t is the input at the current time step, h_t is a hidden state, and f is an activation function.

However, a basic RNN is affected by the long-term dependency problem; therefore, it may be difficult to retain all information in the hidden states over long sequences [14, 22]. [14] proposed long short-term memory network (LSTM) to specifically learn long-term dependency.

At each time step t, we define the LSTM units as vectors in R^d. i_t represents an input gate, c_t represents a memory cell, o_t represents an output gate, f_t represents a forget gate and h_t represents a hidden state. d is the number of units for LSTM. The entries of the gating vectors i_t, f_t, and o_t are between 0 and 1. The LSTM transition equation is as follows.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}) \tag{3}$$
$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right) \tag{4}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t) \tag{5}$$
$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1}) \tag{6}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{7}$$
$$h_t = o_t \odot tanh\ (c_t) \tag{8}$$

where x_t is the input at the current time step, σ is the logistic sigmoid function and $\odot$ is element-wise multiplication. Intuitively, the forget gate controls the amount that is erased from each unit of the memory cell, the input gate controls the updating of each unit, and the output gate controls the exposure of memory content.

Fig. 4 describes the structure of the proposed RNN-LSTM model with unigram embeddings. Since word level embeddings may not be enough to capture the semantic meaning of names, we use character level embeddings. Unigram features are extracted from each letter and input into the RNN-LSTM model. For an initialization of character embeddings, we use Skip-gram as follows [10] did. Details of Skip-gram can be found in [10]. Hidden vectors are passed to additional hidden vectors, which makes the proposed model slightly better. An output of a hidden layer is a softmax probability for predicting an ethnicity.

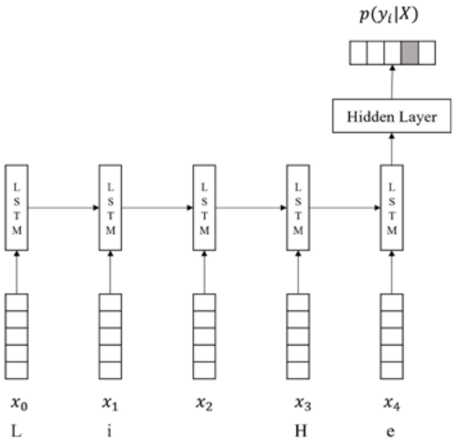


Fig. 4. RNN-LSTM with character-level embeddings

5. Experiments

5.1 Dataset

We apply two approaches (keyword and common surname) to obtain more labeled dataset. Notably, training dataset 1 is manually annotated by a native speaker. Training dataset 2 includes training dataset 1 and additional data which are labeled using the first approach (keyword). As listed in **Table 3**, the ethnicity distribution of training dataset 1 is not balanced (*e.g.*, 382 Chinese names and 7 Indian names). Application of the first approach makes the imbalance ratio between major and minor group more severe because we do not have keywords for certain ethnicities. We exclude some Malay/Muslim group names for under-sampling when we construct training dataset 2 to address this imbalance issue. Similarly, training dataset 3 includes training dataset 2 and additional data which is labeled by taking the second approach (surname). Notably, we assume test datasets does not have any uncertainty because it is manually annotated by a native speaker. Details of training and test data are listed in **Table 3**.

Table 3. Statistics of three train, two test, and additional datasets

Dataset	Malay/Muslim	Chinese	Indian	Western	Other	Total
Train Dataset 1	120	382	7	52	16	577
Train Dataset 2	1176	382	197	52	102	1909
Train Dataset 3	9427	7076	5843	666	118	23130
Test Dataset 1	60	20	20	5	2	107
Test Dataset 2	70	70	50	10	0	200
Additional Dataset	255	298	191	60	0	804

5.2 Computational results

We compare the proposed model with SVM to verify its performance. We measure validation accuracy using k-fold cross validation with k = 5 and test accuracy based on out-of-sample validation to assess the accuracy. Generally, more training data results in better machine learning models; however, we would like to verify if this general statement holds for this case. We consider the benefits of a larger dataset because the label noise level of the training data may increase as the size of training dataset increases.

In **Table 4**, the first and second columns indicate which training data and model we use. The third column indicates the validation accuracy and fourth and fifth columns indicate the test accuracy for two different test datasets. Notably, we randomly split the training dataset with 7:3 ratio for creating validation data. The validation accuracy varies from 0.66 to 0.85 for SVM and 0.81 to 0.89 for RNN based on the training data. The results indicate that the accuracy increases as we have more training data and the proposed model outperforms SVM for all training datasets and improves the accuracy by up to 15%. Because RNN has better performance in capturing features within sequence input, this result is expected.

We verify test accuracy (out-of-sample validation) to evaluate two models more precisely. Additionally, two test datasets are annotated by a native speaker; therefore, we can assume that no label noise exists. The overall test accuracy is decreased compared to the validation accuracy; however, the proposed model outperforms SVM for all test sets. The test accuracy increases monotonically, similar to the validation accuracy as we have more training data. The proposed model exhibits a 9% (on average) better prediction accuracy than SVM.

Table 4. Validation and test accuracy for three different train datasets. Bold numbers indicate better results between two models

Dataset	Model	Val acc	Test acc1	Test acc2
Train Dataset 1	SVM	0.66	0.63	0.52
	Ours (RNN-LSTM)	**0.81**	**0.72**	**0.59**
Train Dataset 2	SVM	0.77	0.63	0.49
	Ours (RNN-LSTM)	**0.83**	**0.72**	**0.62**
Train Dataset 3	SVM	0.85	0.74	0.69
	Ours (RNN-LSTM)	**0.89**	**0.85**	**0.79**

5.3 Transfer learning and fine-tuning

After we deploy a deep learning model in a real-world setting, it is quite common to find that there is additional data available for training. The naive approach is that we train a model with new merged dataset; however, it is quite time-consuming. A better approach would be transfer learning and fine-tuning with the additional dataset. We implement transfer learning and fine-tuning to observe the effect of the additional dataset. Notably, the source and target task are semantically same in our setting and this is not common in the other transfer learning literature.

Table 5. Accuracy change after adding additional train data. Bold numbers indicate better results between two models

Train Data	Model	Test acc1	Test acc2
Train Dataset 1+ Additional Set	SVM	0.74 (+0.11)	0.68 (+0.16)
	Ours (RNN-LSTM)	**0.85 (+0.13)**	**0.73(+0.14)**
Train Dataset 2+Additional Set	SVM	0.71 (+0.08)	0.63 (+0.14)
	Ours (RNN-LSTM)	**0.76 (+0.04)**	**0.72(+0.10)**
Train Dataset 3+Additional Set	SVM	0.79 (+0.05)	0.70 (+0.01)
	Ours (RNN-LSTM)	**0.89 (+0.04)**	**0.81(+0.02)**

We directly use trained parameters from the existing model to initialize the network in the target task and initialize parameters with an additional dataset of 804 samples, as Mou et al. [20] did. We use supervised pretraining; after transfer, we freeze all parameters and train only fully connected layers with additional labeled data for fine-tuning. For fair comparison, we conduct hyperparameter tuning for SVM as follows [23]. **Table 5** shows the test accuracy after transfer learning and fine-tuning with additional dataset. The numbers in parenthesis on the third and fourth columns indicate the improvement with and without using additional data. We observe that using additional dataset is always beneficial and the proposed model outperforms SVM by up to 11%.

5.4 Incorrect prediction analysis

We analyze the validation accuracy of the proposed model with training dataset 3 to investigate its performance more. The prediction results for the validation dataset are presented as confusion matrix in **Fig. 5**. The overall validation accuracy is 0.89. **Fig. 5** shows that the proposed model predicts Indian ethnicity with 0.96 accuracy. However, the model seems to have less accurate predictions for the ethnicity of Malay/Muslim and Chinese although we have more training data for them. Notably, the Indian ethnicity is the most frequent class among incorrect predictions. Specifically, Malay/Muslim group has 312 incorrect predictions and approximately 73% of them are predicted as Indian.

Additionally, for the Chinese group, 240 out of 266 incorrect predictions are predicted as Indian. We attempt to identify some patterns for incorrect predictions; however, we could not discover any clear pattern to verify whether the proposed model is overfitted to Indian ethnicity. Therefore, we will consider this issue in future studies for further improvement of the proposed model. Notably, the Western group is trained well although it has relatively small number of data. This is because Western names are semantically distinguishable from Malay/Muslim and Indian names.

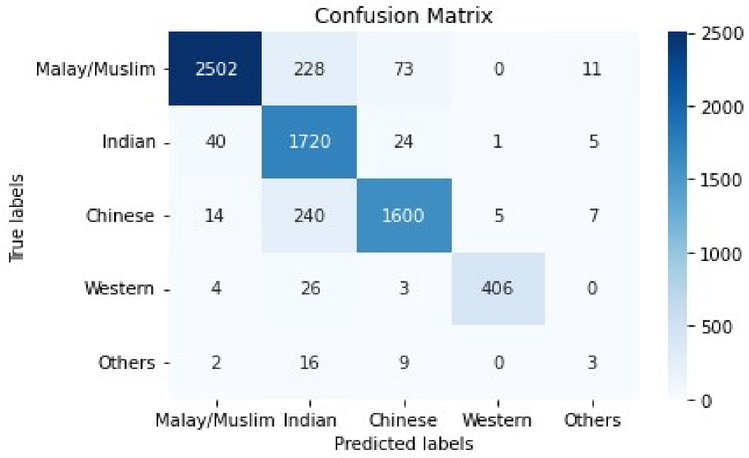


Fig. 5. Confusion matrix using validation dataset

5.5 Separated surname and given name verification

Predicting ethnicity is to apply name-separation rules. However, ultimately, we are unable to evaluate the proposed algorithm without human intervention because we only have labeled dataset which has ethnicity information with full names, not separated surnames and given names. This implies that the final results (*e.g.*, separated surnames and given names by model) must be reviewed by native speakers. We share the final outputs with native speakers and they consist of original full names, predicted surnames, and given names. We ask reviewers to verify whether this separation is correct.

Particularly, we predict the ethnicity of the given full names and apply the name-separation rules based on the (predicted) ethnicity to separate surnames and given names. In an average of three reviewers' feedback, more than 90% of surnames and given names are separated well when the ethnicity prediction is correct. However, we obtain less than 50% of accuracy for separating surnames and given names when we have incorrect ethnicity prediction. This feedback may support the proposed approach that separating surnames and given names based on the ethnicity is effective while struggling with no labeled surname and given name at the beginning.

6. Discussions

In Section 3, we mentioned that 38% of the data are labeled by applying two approaches suggested. This implies that we have a large amount of unlabeled data remaining. In this study, we only consider supervised learning for ethnicity prediction in which we do not utilize any unlabeled data. It will be beneficial to utilize these unlabeled data. Thus, using semi-supervised learning for ethnicity prediction will be considered in future studies. Furthermore, it will be interesting if we can introduce an efficient active learning algorithm that can perform auto-labeling with little human intervention.

7. Conclusion

Malaysia has multiple main ethnic groups and the method for splitting surnames and given names differs based on each ethnicity. Thus, identifying Malaysian surnames and given names is challenging.

First, we proposed two approaches to obtain labeled ethnicity data from scratch and addressed the problem of identifying Malaysian surnames and given names using a two-phase framework. We developed an RNN-LSTM with unigram embeddings for predicting ethnicity using the labeled ethnicity data. Subsequently, we applied a rule-based algorithm for splitting full names into surnames and given names. We evaluated the proposed method with various machine learning algorithms and improved the validation and test accuracy by up to 15 and 13%, respectively. Moreover, we demonstrated that additional dataset can further enhance the prediction accuracy of the proposed model using transfer learning and fine-tuning.

Acknowledgement

This work was supported by INHA UNIVERSITY Research Grant.

References

[1] E. Burchard, E. Ziv, N. Coyle, S. Gomez, H. Tang, A. Karter, J. Mountain, E. P´erez-Stable, D. Sheppard, and N. Risch, “The importance of race and ethnic background in biomedical research and clinical practice,” *New England Journal of Medicine*, vol. 348, no. 12, pp. 1170–1175, Mar. 2003. Article (CrossRef Link)

[2] R.W. Buechley, “Generally useful ethnic search system “GUESS”,” New York, pp. 49–58, 1976.

[3] A.J. Coldman, T. Braun, and R.P. Gallagher, “The classification of ethnic status using name information,” *Journal of Epidemiology and Community Health*, vol. 42, no. 4, pp. 390–395, Dec. 1988. Article (CrossRef Link)

[4] J. Chang, I. Rosenn, L. Backstrom, and C. Marlow, “epluribus: Ethnicity on Social Networks,” *ICWSM*, vol. 4, no.1, pp. 18–25, May. 2010. Article (CrossRef Link)

[5] A. Ambekar, C. Ward, J. Mohammed, S. Male, and S. Skiena, “Name-Ethnicity Classification from Open Sources,” in *Proc. of KDD*, pp. 49–58, Jun. 2009. Article (CrossRef Link)

[6] P. Treeratpituk, and C.L. Giles, “Name-Ethnicity Classification and Ethnicity-Sensitive Name Matching,” in *Proc. of AAAI*, pp. 1141–1147, Jul. 2012. Article (CrossRef Link)

[7] N. Jindal, and B. Liu, “Review spam detection,” in *Proc. of the 16th international conference on World Wide Web. ACM*, pp. 1189–1190, May. 2007. Article (CrossRef Link)

[8] E. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, Feb. 2011. Article (CrossRef Link)

[9] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, “Detecting automation of twitter accounts: Are you a human, bot, or cyborg?,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 811–824, Aug. 2012. Article (CrossRef Link)

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. of the 26th International Conference on Neural Information Processing Systems*, Vol. 2, pp. 3111–3119, Dec. 2013. Article (CrossRef Link)

[11] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014. Article (CrossRef Link)

[12] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994. Article (CrossRef Link)

[13] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014. Article (CrossRef Link)

[14] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. Article (CrossRef Link)

[15] K. Cho, B. Merri¨enboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014. Article (CrossRef Link)

[16] J. Lee, H. Kim, M. Ko, D. Choi, J. Choi, and J. Kang, "Name Nationality Classification with Recurrent Neural Networks," in *Proc. of International Joint Conference of Artificial Intelligence Organization*, pp. 2081–2087, 2017. Article (CrossRef Link)

[17] Y. Yao, and Z. Huang, "Bi-directional LSTM Recurrent Neural Network for Chinese Word Segmentation," in *Proc. of International Conference on Neural Information Processing*, pp. 345–353, Feb. 2016. Article (CrossRef Link)

[18] A. Dai, and Q. Le, "Semi-supervised sequence learning," in *Proc. of the 28th International Conference on Neural Information Processing Systems*, Vol. 2, pp. 3079–3087, Dec. 2015. Article (CrossRef Link)

[19] J. Howard, and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistic*, pp. 328–339, Jul. 2018. Article (CrossRef Link)

[20] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin, "How transferable are neural networks in NLP applications?," in *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 479–489, Nov. 2016. Article (CrossRef Link)

[21] S. Min, M. Seo, and H. Hajishirzi, "Question Answering through Transfer Learning from Large Fine-grained Supervision Data," in *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pp. 510–517, Jul. 2017. Article (CrossRef Link)

[22] John F. Kolen, Stefan C. Kremer, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," *A Field to Guide to Dynamical Recurrent Neural Networks*, pp. 237–243, 2001. Article (CrossRef Link)

[23] A. Karatzoglou, D. Meyer, and K. Hornik, "Support vector machines in R," *Journal of statistical software*, vol. 15, no. 9, pp. 1–28, 2006. Article (CrossRef Link)

[24] V. Selvaratnam, "Ethnicity, inequality, and higher education in Malaysia," *Comparative Education Review*, vol. 32, no. 2, pp. 173–196, 1988. Article (CrossRef Link)

[25] J. Kim, J. Kim, and J. Smith, "Ethnicity-based name partitioning for author name disambiguation using supervised machine learning," *Journal of the Association for Information Science & Technology*, vol. 72, no. 8, pp. 979–994, August 2021. Article (CrossRef Link)

[26] K. Wong, O. Zaiane, F. Davis, and Y. Yasui, "A machine learning approach to predict ethnicity using personal name and census location in Canada," *PLoS ONE*, vol. 15, no. 11, 2020. Article (CrossRef Link)

YOUNGBUM HUR received the Ph.D. degree in operations research and industrial engineering from The University of Texas at Austin, in 2017, under the supervision of Jonathan F. Bard. He worked at Sabre, from 2017 to 2019, and the Samsung Advanced Institute of Technology (SAIT), from 2019 to 2021. He is currently an Assistant Professor with the Department of Industrial Engineering, Inha University. His early work are mostly on operations research. After he graduated, he started to work on machine learning and deep learning.