Finding the Gender of Personal Names and Finding the Effect of Gana on Personal Names with Long Short Term Memory

T.C. Lekamge[#1] and T.G.I. Fernando[#2]
[#]*Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka,10250*
[1]thisuri.lekamge94@gmail.com
[2]gishantha@dscs.sjp.ac.lk

***Abstract* - Naming a baby is a very integral part of our life. As well as, naming a business with an attractive and suitable name takes the first place in business world. In Asian countries, especially in Sri Lanka, people tend to follow rituals and customs introduced by astrology when naming a baby or a business. Because they believe that good luck can be achieved by following astrological concepts like 'Mathra' and 'Gana' in naming. As a result of that, people tend to spend more money in finding a suitable name which follows astrological concepts. Also, when naming a new born according to gender, parents think whether that name is suitable for their baby boy/girl or not. On the other hand, as web services grow, personal names have become one of the most abundant sources of data on the web. So, if one can predict or suggest proper gender for personal names; web or mobile applications can benefit from knowing each user's gender. Research studies on predicting the gender of personal names and the effect of 'Gana' on personal names with deep learning algorithms have not been studied. Therefore, our main goal was to implement separate models for gender classification and finding the effect of 'Gana' on personal names (forenames/most use name) which are in text format using Recurrent Neural Networks. As a result of that, we developed two LSTMs (a special type of Recurrent Neural Networks) that produced promising results.**

Keywords: Machine Learning (ML), Long Short Term Memory (LSTM), Astrology, Natural Language Processing (NLP)

I. INTRODUCTION

In the last few centuries modern science has changed the life of human being in an unprecedented way. Science has not only changed our life style but also the attitudes and beliefs that have lasted more than thousands of years in human history. Some of those ancient beliefs have survived the waves of modern science and still has the control over day to day life of humans. Astrology is such an ancient science [20]. Astrology has been affecting the life of Asian people for centuries. Therefore, Sri Lankans, specially Sinhala Buddhists and Hindus follow customs and rituals introduced by astrology.

So there are many astrological parameters and measures to ascertain a person's future and character. These rituals start with the birth of a new born and ends with the time of his death. First of these rituals is naming a baby. At present many parents tend to find names for their children following astrological concepts and spend lot money on this matter. Also, by today, when a business is started, owners tend to find a better name for their business following astrological concepts. It is believed that ancient Sinhalese had the capability of narrating poems (සෙත් කවි/වස් කවි) which were able to bring good or bad luck on people. The secret behind these poems are astrological phonology [20], [26]-[28].

Therefore, astrological phonology is widely used today in suggesting names for new born children and businesses. There are many concepts in astrological phonology [20]. Two of these concepts are mainly followed in naming. 'Mathra' (මාත්‍ර) is one of them. 'Mathra' is considered as the measurement of time taken for pronouncing a word in astrology. Traditional astrologers consider the time taken for a blink is equal to a 'Mathra.' This measurement is not scientific but when considering the examples given in various texts it is clear that a 'Mathra' is the time taken to pronounce a single short vowel or a consonant joined with a short vowel. For example: අ /a/ is a short vowel which takes the time of a 'Mathra' to be pronounced. ක /ka/ is a consonant /k/ joined with the short vowel /a/ which takes the time of a 'Mathra' to be pronounced. In naming, Short Mathra (අ /a/) and Long Mathra (ආ /à/) are used.

Other one is 'Eight Gana' (අෂ්ඨ ගණ) which is directly used in naming concepts. 'Gana' is formed by the time taken for a word to be pronounced or how the sounds within the word are grouped together when it is pronounced. This is used in predicting the effect of a word. Without knowing 'Mathra' of a word, finding 'Gana' is impossible. Because of that, two concepts; 'Mathra' and 'Gana' are connected to each other. There are eight kinds of 'Gana' [20], [26]-[28] and astrologers have given effects of these 'Gana' as follows.
(a = Short Mathra, c = Long Mathra)

TABLE I
GANA WHICH GIVE GOOD LUCK

Name of Gana	Sequence of Mathra	Effect
Land	ccc	Wining
Moon	aaa	Life
Deva	caa	Fame
Water	acc	Development

TABLE II
GANA WHICH GIVE BAD LUCK

Name of Gana	Sequence of Mathra	Effect
Sun	aca	Illness
Fire	cac	Destruction
Air	aac	Death
Celestial	cca	Losses

Predicting the effect of 'Gana' on names using deep learning algorithms has not been studied previously. Deep learning is becoming a fast developing area in the field of computer science. Also Natural Language Processing

978-1-7281-5156-4/19/$31.00 ©2019 IEEE

(NLP), Text Mining and Sequence Learning are trending areas in deep learning. Therefore, in our research study, deep learning has been applied to this astrological problem using Recurrent Neural Networks (RNNs).

Our research study consists of two tasks; one of them is finding the effect of 'Gana' on personal names. The main interest of this task is the astrological phonology. For that, above mentioned astrological concepts were studied and a Long Short Term Memory (LSTM) based model which predicts the effect of 'Gana' of a name/word was developed. The importance of our research is that personal names are inputted as text. Therefore, we had to preprocess our own datasets according to our own phonological alphabet which is compatible with astrological phonology too. Also here we could train our model for names from different countries with a high accuracy.

Apart from that, naming a baby is not only connected with astrology but also with the language. Actually, a name is a term used for identification. Names can identify a class or category of things, or a single thing, either uniquely, or within a given context. Among them, a personal name identifies a specific person and categorizes that person as male or female, since gender plays a fundamental role in social interactions. Also this gender classification based on personal names is mainly connected with regional languages. Therefore, these personal names can be used to investigate how characters are arranged for naming according to that language. Among proper nouns, personal names tend to have many conventional structures than other proper nouns such as company names or acronyms. These structures are clearer when the names are confined to a specific gender. For instance, if we consider Sri Lankan forenames/ first names, we can see last letter often indicates the gender of first names. It is needed to be considered that these structures differ according to regional languages. For instances, Sri Lankan feminine forenames tend to have "i" or "e" at the end (e.g., Nilmini, Thisuri, Nipuni, Panjalee). Sri Lankan masculine forenames tend to have "m", "l", "n", "h" or "u" at the end (e.g. Kelum, Lakmal, Shaveen, Jagath, Isuru). But is this always correct? There are some masculine forenames ending with "i" (e.g. Gamini) though the letter "i" is reserved for feminine forenames. Also the letter "a" is included in both masculine (e.g. Nadeera) and feminine forenames (e.g. Nayana (Nayanà)). Due to this diversity and subtlety, naming conventions are quite difficult to distinguish.

On the other hand, as web services grow, personal names have become one of the most abundant sources of data on the web [1]. If one can predict or suggest proper gender for personal names; services such as web applications or mobile applications can be benefitted from knowing each user's gender.

Normally, predicting gender based solely on personal names using deep learning algorithms has been poorly studied due to its difficulties in finding subtle structures between naming conventions. In our research study, as the other task, an LSTM based model which can predict gender based on Sri Lankan personal names was developed with a high accuracy.

The main objective of this research study was to develop two RNN based models with high accuracies, one for gender classification and other for finding the effect of 'Gana' on personal names. Therefore, this paper explores the implementations and the results of those developed models later.

II. RELATED WORK

A. Personal Name Classification

Liu and Ruths [2] focused on gender classification in Twitter using SVM-based classifier and first names. Thus, much work has been done on the problem of gender inference on Twitter. The top performing methods in this area use feature-based classifiers such as support vector machines and boosted decision trees [3, 4]. Probabilistic models such as Naïve Bayes and latent semantic analysis have also been considered. Pennacchiotti and Popescu [4] classified Twitter users into nationalities or ethnicities using several machine learning (ML) approaches. They used various features such as name, profile, tweeting behavior and linguistic content. Another approach of Twitter user classification is studied by Bergsma et al. [5]. Instead of utilizing user profiles, they used clusters of users' first names, last names and locations to identify users. Huang et al. [6] utilized Twitter based ethnicity classification results and other information such as user profiles to predict a user's nationality. Chang et al. [1] trained a classifier using a Bayesian approach with U.S. Census name data. The classifier found relationships between Facebook user names and ethnicities. We can see, above name classification models have been used for classifying users in social networks.

Ethnicity classification is one of the main tasks that utilizes personal names. Ambekar et al. [7] classified 13 cultural groups using a decision tree and a Hidden Markov Model on a news corpus. Jinhuyak et al. [8] classified nationality of each personal name using a recurrent neural network based model and utilized skip-gram based embedding. Humphreys et al. [9] used the ethnicity classification method to find out the relationship between house marketing and ethnicity and prove the cultural superstitions of Chinese.

B. Gender Classification

As described in above section *A*, much work has been done on the problem of gender inference on Twitter. Apart from that, the problem of automatically extracting gender related attributes from facial images has been received increasing attention in recent years. Gil et al. [16] conducted age and gender classifications with convolutional neural network using facial images.

There are well-known linguistic differences between the writing of men and women. Using this concept Deitrick et al. [17] performed a gender prediction in an email stream using modified balanced Winnow Neural Network. Noah et al. [19] introduced a deep learning model for gender identification, which captures contextual information of each word in the texts and interprets in the left-to right manner of the text. Reza et al. [18] used pitch feature of voice for classification between males and females. Their method is based on Multi-Layer Perceptron Neural Network.

C. Finding effect of Gana

Research studies on finding the effect of 'Gana'/Astrology on personal names using machine learning approaches have not been studied to our knowledge and the literature we studied. Therefore, our study is the first attempt to apply deep learning to this astrological problem.

D. Recurrent Neural Networks (RNNs)

RNNs are known for their ability to predict sequential data such as natural languages. Mikolov et al. [10] showed that RNN is effective on language modeling. Bahdanau et al. [11] used RNNs for machine translation and they achieved performances comparable with statistical machine translation models. Due to its recurrent structure, recurrent neural networks tend to suffer from long-term dependency [12] and severe overfitting problems [13]. To learn long-term dependencies, some researchers suggested LSTM which significantly reduces the long term dependency problem using memory cell and forget gate. A similar RNN cell, Gated Recurrent Unit (GRU) was introduced to improve the efficiency of LSTM. Overfitting problems of RNNs are alleviated with applying dropout on non-recurrent connections of RNNs [13].

E. Drawbacks of earlier research studies

As discussed in section *A*, many researchers have used number of features for classification task. For instance, sometimes, gender classification in Twitter was based on users' first name, last name, tweets, writing style etc. But here, this research is targeted at developing the models using only forenames/most-used names.

III. "METHODOLOGY

A. Dataset

Here we need two datasets for gender classification and finding the effect of 'Gana.' In these two tasks we used personal names. It is to be noticed that full name has not been used here; we restricted it to one name which is used as forename or most-used name by a particular person. Since there are no gender-labeled or Gana-effect-labeled personal name datasets available to the research community, we created ours to use in this study.

1) Data Collection: Names were collected from websites, social media, papers and people. As mentioned before, two name datasets were created. Also names were in English; e.g. අයනි -> Ayani.

For the gender classification, first we collected data under three categories (male, female and unisex) and further we categorized it into two (including unisex names under both male and female categories); those are,

Male	1,581
Female	1,531
Total	3,112

Here 'Male' means personal names (as described in *A*) used by male persons, as well as 'Female' means personal names used by female persons. Also this dataset is mainly consisted with Sri Lankan personal names.

For finding the effect of 'Gana,' we categorized the collected data into three; those are,

Good	1,195
Bad	1,230
Neutral	1,825
Total	4,250

Here 'Good' means if the 'Gana' of a name is good, and therefore the effect of the name supposed to be good. As well as, 'Bad' means 'Gana' of the name is bad and therefore effect of the name is bad to the personal life. Apart from those, there are some names which we cannot find 'Gana' since there are no at least three sounds in that particular name. Those kinds of names are called as neutral names by astrologers. Also they believe neutral names give neutral or both good and bad results to personal lives. In that manner, those names have been categorized under neutral. In this dataset, different names from different countries were included.

Apart from that, we always tried to create 'balanced' datasets. Balanced dataset is a dataset which has equal number of instances in each class. Imbalanced data typically refers to a problem with classification problems where the classes are not represented equally. For an instance; in a binary classification problem with 100 instances, a total of 80 instances are labeled with Class-1 and the remaining 20 instances are labeled with Class-2. This is an imbalanced dataset. Because of highly unbalanced dataset, frustrating results can be caused. But in our datasets, exactly equal numbers of instances are not in each class, but a small difference often does not matter.

2) Data Preprocessing: This study is focused on two tasks: gender classification and finding the effect of 'Gana.' So these tasks are mainly based on the pronunciation of names.

There are some Sri Lankan names written in same pattern in English but pronunciation is different according to gender. For instances, Male name 'අකිල', we write in English as 'Akila' as well as Female name 'අකිලා' also, we write in same manner as 'Akila.' For these kinds of names, we introduced a writing pattern. Actually, this is specially affected to female names which end with 'long a' sound (á), e.g. Akilá, Nayaná. But in our normal usage, we do not use letters with acute (á) for long sounds. In our introduced writing pattern (our own phonological alphabet as shown in Fig. 1), we follow easy patterns to replace those kinds of letters. Especially in this task, those ending 'a's in female names were replaced with 'aa.' Also all long sound letters were replaced as in Fig. 1. For an instances, Akilá as Akilaa, and Nayaná as Nayanaa have been written.

In finding the effect of 'Gana,' pronunciation is an integral part. According to Astrology, Gana of a name is mainly based on how one pronounces that particular name. Most of us do not write our names as in pronunciation. So here we introduced a particular writing pattern (our own phonological alphabet) to write our name as in pronunciation.

In this manner, to train our models for different pronunciations, we preprocessed both datasets according to this writing pattern as shown in Fig. 1. Otherwise results

can be different than expected. Apart from that, preprocessing was done to remove duplicates.

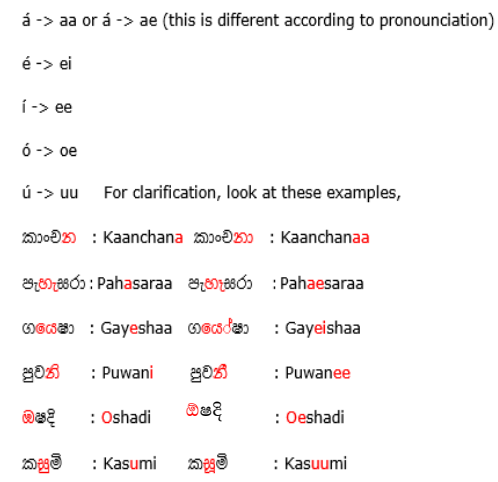

Fig.1 Introduced writing pattern for letters with acute (our own phonological alphabet)

B. Algorithms

RNN is a good fit for this as it involves learning from sequences (in these cases sequences of characters). But traditional RNN has learning problems due to vanishing gradients. RNN have shown two variants that can help to solve this problem, LSTM is one of them.

We selected LSTM. Meanwhile we did our study with Simple Neural Network to compare the performances between them. Eventually, LSTM has been selected as its performance is superior to the vanilla neural network.

C. Tools and Technologies

Deep learning is consisting of large number of calculations. Therefore, usage of high end hardware is a necessary. Most novel deep learning libraries are supporting GPU computations for making deep learning calculations faster. In this study, a GPU was used for the training phase of our deep networks and libraries which support GPU implementation were used.

1) System hardware and platform: For this study, a personal computer was used with Intel Core i7-8700 CPU and GeForce GTX 1060 GPU. Network training phase was conducted on a GPU. Main reason for using a GPU is the fast computation time of deep learning implementations. Ubuntu 18.04.1 LTS which is 64 bit operating system, as selected as our implementation platform. Ubuntu is a Linux platform. Most of the deep learning tools were fully tested on Linux based platforms with a good working quality. Apart from that, Linux systems also have a great support for GPU computation libraries.

2) Machine Learning libraries and frameworks: Mainly 'Keras' with 'Tensorflow' backend was used in this study. One of the most powerful and easy to use Python libraries for developing and evaluating deep learning models is 'Keras.' It wraps the efficient numerical computation libraries, 'Theano' and 'Tensorflow.' The advantage of this framework is mainly we can get started with neural networks in an easy way. Also network architecture can be changed in efficient manner with fewer changes in codes because of its flexibility. This provides a fast and easy prototyping of network with support of multi-input and multi-output training. 'Keras' allows user to change network hyper parameters more easily.

'Tensorflow' is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs) and from desktops to clusters of servers to mobile and edge devices.

D. Implementation

Because of this study is based on sequence learning, RNN was used without any hesitation and also as the deep learning primarily step, we implemented a simple neural network architecture to compare the performances. Since the literature regarding gender classification and finding the effect of 'Gana' on personal names with recurrent neural network was not available, we developed our own models with many experiments. In that manner, two models were developed for gender classification and finding the effect of 'Gana' respectively.

1) Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM): RNN can process temporal data and sequential data. RNNs allow the neural network to understand the context in speech, text or music. An RNN contains at least one feedback connection, so the activation can flow round in a loop. That is, they have a memory which captures information about what has been calculated before. According to that, an RNN can process an arbitrary long sequence, however in practice they are limited to looking back only to a few steps. Actually the main feature of an RNN is, its hidden state which captures some information about a sequence.

But training RNNs with stochastic gradient descent to observe long term dependencies is often challenging. This is because the magnitudes of the gradients tend to vary greatly, researchers observed that during back propagation. When this happens in a neuron, it is saturated and drives gradients in previous layers towards zero. Thus small gradient values lead multiple multiplications of gradients shrinking exponentially fast and eventually vanishing completely after a few steps. As a result of that, gradient contribution from 'faraway' steps become zero and the states of those steps do not contribute the learning and end up with no learning of long-range dependencies.

In order to handle the vanishing gradient problem, more sophisticated recurrent units have been proposed. Long Short Term Memory (LSTM) unit is one of them, which was proposed by Sepp Hochrieter and Juergen Schmidhuber [22] in 1997. LSTM widely use in Natural Language Processing tasks. LSTM was designed to combat vanishing gradients through a 'gating' mechanism. This special unit can learn to bridge time intervals in excess of 1,000 steps. All RNNs consist of chain of repeating modules of neural networks. In a vanilla RNN, this repeating module has a simple structure, such as a single 'tanh' layer. LSTMs also have this chain like structure, but

the inner structure is different from an RNN. Instead of having a single layer, there are four, increasing in a very special way.

The original LSTM model is comprised of a single hidden LSTM layer followed by a standard feed forward output layer. But we have used stacked LSTM models in our study. The stacked LSTM is an extension to the original LSTM model that has multiple hidden layers where each layer contains multiple memory cells. In our gender classification model, we used three LSTM layers as shown in Fig. 2.

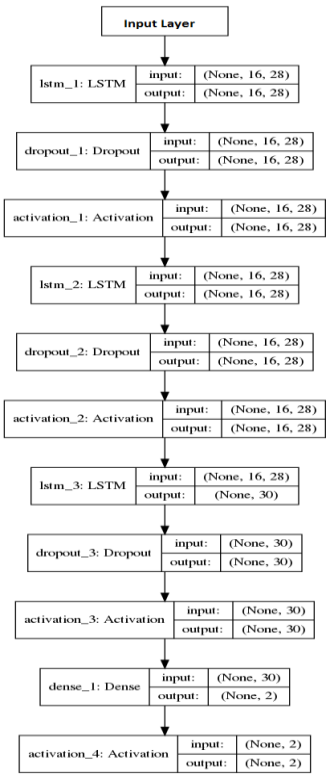

Fig. 2 Model architecture for gender classification

In finding the effect of 'Gana' model, two LSTM layers were used as shown in Fig. 3.

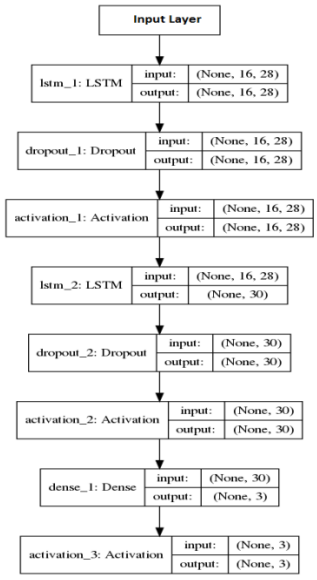

Fig. 3 Model architecture for finding the effect of 'Gana'

Stacking LSTM makes the model deeper. It is the depth of neural networks that is generally attributed to the success of the approach on a wide range of challenging prediction problems. Therefore, stacked LSTMs are now a stable technique for challenging sequence predictions problems.

2) Input Parameters: Vocabulary was 28 chars including 'a-z' (both upper case and lower case letters are considered as lower case letters), space and a special END token. Max sequence length was chosen as 16. If a name has less than 16 characters 'END' token is padded.

3) One Hot Encoding approach: One hot encoding is a representation of categorical variables as binary vectors that could be provided to ML algorithms to do a better job in prediction. The first requires that categorical values be mapped to integer values. Then each integer is mapped as a binary vector that is all zero values except the index of the integer. In our study, each character is one hot encoded.

4) Softmax activation function: Softmax activation function is an extension of the logistic regression function which calculates the probability of the input belonging to every one of the existing classes. The class with the highest probability is chosen as the predicted class. This function squashes an n-dimensional vector of arbitrary real values to an n-dimensional vector of real values in the range (0,1) that add up to 1. Softmax activation is generally used at the output layer to get probabilities as it pushes the values between 0 and 1. As it is we used softmax activation function with LSTM layers and the dense layer at the end. Dense layer was added with required dimensionality to get the desired target. First we used sigmoid activation function in intermediate layers, but after using softmax activation function in all layers, there was an improvement in accuracy of the models.

5) Dropout layers: Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Regularization is a great approach to curb overfitting the training data. Dropout is a hot new regularization technique for addressing this problem. The key idea is to randomly drop units along with their connections from the neural network during the training process. This makes sure the network is not getting too 'fitted' to the training data and thus helps alleviate the overfitting problem. Generally, maximum dropout probabilities for hidden units are in range 0.5 to 0.8. During our study we tested our models with different dropout probabilities. In that manner, 0.1 and 0.3 were used as dropout probabilities in gender classification model and 0.1, 0.5 and 0.7 were used as dropout probabilities in finding the effect of 'Gana' model.

6) ADAM Optimizer: This can be used instead of stochastic gradient descent optimization methods to iteratively adjust network weights. Adam [21] is computationally efficient, works well with large data sets, and requires little hyperparameter tuning, according to researchers Adam uses an adaptive learning rate α, instead of a predefined and fixed learning rate. In practice Adam is currently recommended as the default algorithm to use, and

often works slightly better than RMSProp [23]. In this study, we previously used RMSProp. Then we moved to Adam optimizer and got better results than previous.

7) Cross Entropy Loss Function: For solving various classification problems, a family of appropriate loss functions has been formulated. They try to maximize the likelihood of the correct class. Maximum Likelihood Estimation (MLE) essentially boils down to maximizing the Cross Entropy (CE) between the empirical data distribution of the labels p and the predicted model distribution q. This is why they are called Cross Entropy Loss or sometimes, due its logarithmic structure, also Log Loss.

In our research study, gender classification is a binary classification; therefore, we used binary cross entropy. Also, finding the effect of 'Gana' is a multiclass classification, therefore categorical cross entropy was used.

8) Batches and Epochs: The batch size defines the gradient and how often to update weights. An epoch is the entire training data exposed to the network, batch-by-batch. Here we experimented with different batch sizes and number of epochs since LSTMs are quite sensitive to batch size and epoch size.

IV. "RESULTS AND DISCUSSION

Here we discuss the results that have been obtained for the implemented solutions for gender classification and finding the effect of 'Gana' on personal name data. In that manner, this includes time consumption for training both models, accuracy and loss in graphs and so on. In this study, the famous accuracy measurements known as accuracy, precision, recall and F1 score were used as the accuracy measurements.

A. Gender classification

Here we implemented our model as in Fig. 2. This model took 20 minutes to train. The final accuracy was 94.94% approximately 95% with 1,000 epochs. As mentioned in implementation, we compared performances using simple neural network architecture. Then we got accuracy and loss as follows.

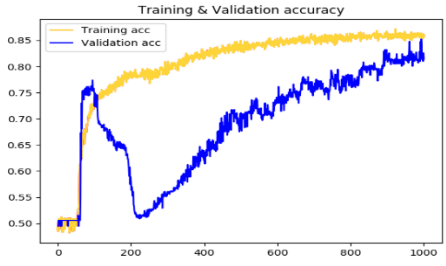

Fig. 4 Accuracy plot for simple neural network architecture

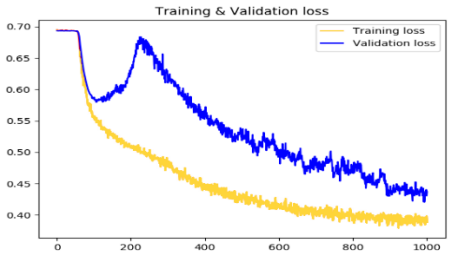

Fig. 5 Loss plot for simple neural network architecture

Here, using a simple neural network architecture we could not get good performance and good accuracy. To get a good accuracy we had to increase number of layers and other parameters, but it is not a good approach in deep learning since it can cause the network complexity. Therefore, LSTM was chosen as mentioned above and accuracy and loss plots were as follows.

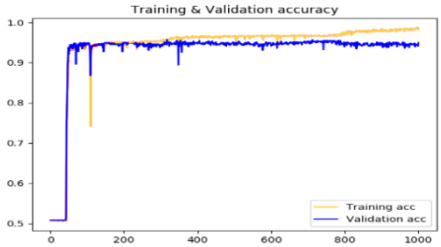

Fig. 6 Accuracy plot for LSTM model

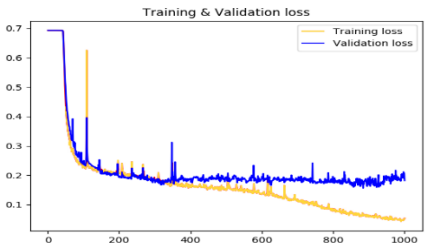

Fig. 7 Loss plot for LSTM model

More statistics such as precision, recall, F1 score and confusion matrix are shown in Table III, Table IV and Fig. 8.

TABLE III
PRECISION, RECALL AND F1 SCORE FOR GENDER CLASSIFICATION

	Female	**Male**
Precision	0.968	0.9651
Recall	0.9645	0.9685
F1 Score	0.9662	0.9668
Support	282	286

Actually, we can see higher measurements for both, female and male. Also, from these statistics we can say that model seems to have learnt patterns in sequences very well. Looking at the predictions also, we can say that complex patterns have been picked up by the model which is not easily visible through inspections. Confusion matrix depicts it very well.

TABLE IV
CONFUSION MATRIX FOR GENDER CLASSIFICATION

	Female	Male
Female	272	10
Male	9	277

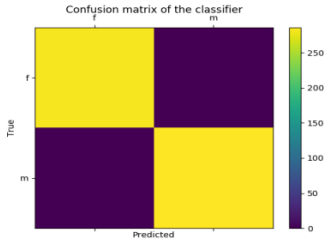

Fig. 8 Confusion matrix for gender classification

This means, 19 names (very little number of samples) were incorrectly predicted out of the total of 568 samples.

B. Finding the effect of 'Gana'

Here we implemented our model as in Fig. 3. This model took approximately 20 minutes to train. The final accuracy was 95.4% with 1,500 epochs. As mentioned in implementation, we compared performances using a simple neural network architecture. Then we got accuracy and loss as follows.

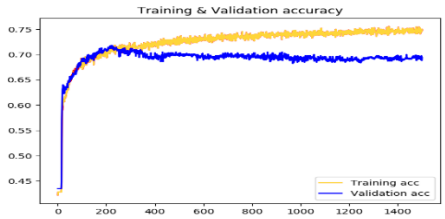

Fig. 9 Accuracy plot for simple neural network architecture

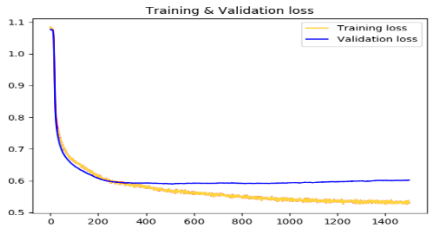

Fig. 10 Loss plot for simple neural network architecture

Here, to get a good accuracy we had to increase the number of layers and other parameters, but it is not a good practice in deep learning since it can cause the network complexity. Therefore, we chose LSTM as mentioned above and accuracy and loss plots were as follows.

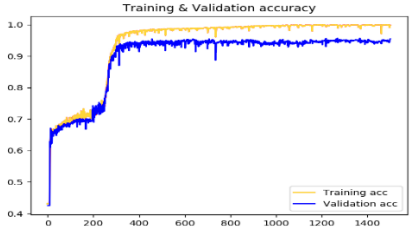

Fig. 11 Accuracy plot for LSTM model

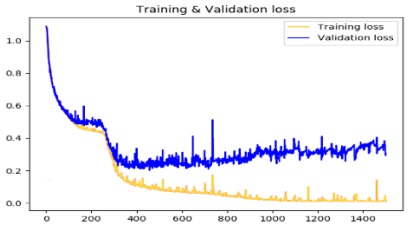

Fig. 12 Loss plot for LSTM model

More statistics such as precision, recall, F1 score and confusion matrix are shown in Table V, Table VI and Fig. 13.

TABLE V
PRECISION, RECALL AND F1 SCORE FOR FINDING THE EFFECT OF 'GANA'

	Bad	Good	Neutral
Precision	0.988	0.988	0.989
Recall	0.980	0.992	0.992
F1 Score	0.984	0.990	0.990
Support	253	251	381

We can see higher measurements for Bad, Good and Neutral. Also, from these statistics we can say that model seems to have learnt patterns in sequences very well. F1 score tells how precise classifier this is. Confusion matrix proves it very well.

TABLE VI
CONFUSION MATRIX FOR FINDING THE EFFECT OF 'GANA'

	Bad	Good	Neutral
Bad	249	1	1
Good	2	248	3
Neutral	1	2	378

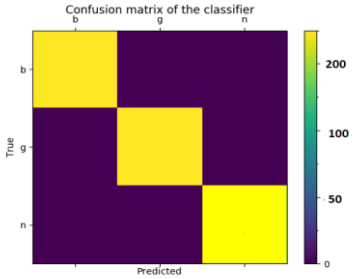

Fig. 13 Confusion matrix for finding the effect of 'Gana'

This means, 10 names (very little number of samples) were incorrectly predicted out of the total of 885 samples.

We can clearly observe that LSTMs show the best performances for two problems discussed in this paper.

V. ⁿ CONCLUSION AND FUTURE WORK

In this study, we implemented a three-stacked LSTM layer model for gender classification and two-stacked LSTM layer model for finding the effect of 'Gana.' Specially, we were able to test these two models with good accuracies. Therefore, we can conclude that LSTM (a special unit of RNN) is a suitable approach for gender

classification and finding the effect of 'Gana' on personal names.

In our study, we were able to get high accuracies with correct predictions using the embedding approach; one-hot encoding. By today, latest embedding approach is Word Embedding. Word embedding is a set of language modeling and feature learning techniques in Natural Language Processing where words or phrases from the vocabulary are mapped to vectors of real numbers. There are two specific forms of word embeddings; Word2Vec [24] and GLoVe [25], collectively known as distributed representations of words. Pre-trained word embedding vectors are also available now. In our study, this technique can be applied to characters. As a solution pre-trained character embedding can be used. Normally, pre-trained character embedding vectors for personal names are not available. Therefore, if we need to use them, we have to create our own.

In our study, personal name data set was limited to Sri Lankan names for gender classification task and therefore, one-hot encoding approach performed well. Further, we can expand our data-set for different names from different countries for gender classification, and then our model need to be trained for more different and complex naming conventions than in this study. For that, using character embedding approach will be better than using one-hot encoding approach.

REFERENCES

[1] Jonahan Chang, Itamar Rosenn, Lars Backstrom and Cameron Marlow. epluribus: *Ethnicity on social networks*. ICWSM, 10:18-25, 2010.

[2] Wendy Liu and Derek Ruths. *What's in a name? using first names as features for gender inference in twitter*. In AAAI spring symposium: Analyzing microtext, volume 13, page 01, 2013.

[3] Burger J., Henderson J., and Zarrella. *Discriminating gender on Twitter*. In proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011.

[4] Pennacchiotti and Popescu. *A machine learning approach to twitter user classification*. In proceedings of the International Conference on Weblogs and Social Media, 2011.

[5] Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson and David Yarowsky. *Broadly Improving user classification via communication based and location clustering on twitter*. In HLTNAACL, pages 1010-1019, 2013.

[6] Wenyl Huang, Ingmar Weber and Sarah Vieweg. *Inferring nationalities of twitter users and studying Inter-national linking*. In Proceedings of the 25th ACM conference on Hypertext and social media, pages 237-242. ACM, 2014.

[7] Anurag Ambekar, Charles Ward, Jahangir Mohammed, Swapna Male and Steven Skiena. *Name-ethnicity classification from open sources*. In proceedings of the 15th ACM SIGKDD International conference on Knowledge Discovery and Data Mining, pages 49-58. ACM, 2009.

[8] Jinhyuk Lee, Hyunjae Kim, Miyoung Ko, Donghee Choi, Jaehoon Choi and Jaewoo Kang. *Name Nationality Classification with Recurrent Neural Networks*. In proceedings of the 26 th International Joint Conference on Artificial Intelligence, 2017.

[9] Brad R Humphreys, Adam Nowak and Yang Zhou. *Cultural superstitions and residential real estate prices: Transaction level evidence from the US housing market*, 2016.

[10] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernock`y, and Sanjeev Khudanpur. *Recurrent neural network based language model*. In Interspeech, volume 2, page 3, 2010.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural machine translation by jointly learning to align and translate*. arXiv preprint arXiv:1409.0473, 2014.

[12] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks, 5(2):157–166, 1994.

[13] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. *Recurrent neural network regularization*. arXiv preprint arXiv:1409.2329, 2014.

[14] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. *Natural language processing (almost) from scratch*. Journal of Machine Learning Research, 12(Aug):2493–2537, 2011.

[15] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. *Character-aware neural language models*. arXiv preprint arXiv:1508.06615, 2015.

[16] Gil Levil and Tal Hassner. *Age and Gender Classification using Convolutional Neural Networks.* CVPR15, 2015.

[17] William Deitrick, Zachary Miller, Benjamin Valyou, BriaDickinson, Timothy Munson and Wei Hu. *Author Gender Prediction in an Email Stream Using Neural Networks*. Journal of Intelligent Learning Systems and Applications, 4(Aug):169-175, 2012.

[18] Mostafa Rahimi Azghadi, Reza Bonyadi and Hamed Shahhosseini. *Gender Classification Based on FeedForward Backpropagation Neural Network*. In IFIP International Federation for Information Processing, volume 247, 2007.

[19] Noah Fleming and Alex Edmonds. *Machine Learning Methods for Gender Identification*.

[20] R.M.E.J. Rathnayaka, *Development of a Tool to Find the Astrological Effect of a Name.* B.Sc. (Hons) degree in Computer Science dissertation, Department of Computer Science, University of Sri Jayewardenepura, Sri Lanka, Apr. 2015.

[21] Diederik Kingma and Jimmy Ba. *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.

[22] S Hochreiter and J Schmidhuber. *Long short-term memory*. Neural computation 9 (8), 1735-1780, 1997.

[23] Sebastian Ruder. *An overview of gradient descent optimization algorithms.* arXiv:1609.04747v2, 2017.

[24] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. *Efficient Esimation of Word Representation in Vector Space.* arXiv: 1301.3781v3 [cs.CL], 2013.

[25] Jeffrey Pennington, Richard Socher and Christopher D. Manning. *GloVe: Global Vectors for Word Representation*. In proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014.

[26] Udendras. (2018). *උදනිදුගේ සිතුවිලි*] .online] Available at: https://udendras.blogspot.com/ [Accessed 7 Mar. 2018].

[27] Vidyaratne, V. (2012). *සුබ නමකින් දරුවාට යහපතක් වන බව සැබෑවක්*] .online] Available at: https://www.sinhalamag.com/2012/02/blog-post_24.html [Accessed 6 Mar. 2018].

[28] Anon, (2014). *බබාට නමක්*. [online] Available at: http://sinhala-subs.blogspot.com/p/blog-page_24.html [Accessed 8 Apr. 2018].