Name Nationality Classification with Recurrent Neural Networks

Jinhyuk Lee[†], Hyunjae Kim[‡], Miyoung Ko[†], Donghee Choi[§†], Jaehoon Choi[§], Jaewoo Kang[†]
[†]Korea University
[‡]Sogang University
[§]Konolabs, Inc.
jinhyuk_lee@korea.ac.kr, gamica@sogang.ac.kr, gomi1503@korea.ac.kr,
choidonghee@korea.ac.kr, jchoi@kono.ai, kangj@korea.ac.kr

Abstract

Personal names tend to have many variations differing from country to country. Though there exists a large amount of personal names on the Web, nationality prediction solely based on names has not been fully studied due to its difficulties in extracting subtle character level features. We propose a recurrent neural network based model which predicts nationalities of each name using automatic feature extraction. Evaluation of Olympic record data shows that our model achieves greater accuracy than previous feature based approaches in nationality prediction tasks. We also evaluate our proposed model and baseline models on name ethnicity classification task, again achieving better or comparable performances. We further investigate the effectiveness of character embeddings used in our proposed model.

1 Introduction

Personal names can be used to investigate how characters are arranged for naming. Among proper nouns, personal names tend to have many more conventional structures than other proper nouns such as company names or acronyms. These structures become clearer when the names are confined to a specific ethnicity. For instance, Russian names tend to have "-sky" or "-v" at the end (e.g., Melnitsky, Shagaev), and German names frequently include "-mann" or "-urg-" (e.g., Hermann, Burgher). However, these naming conventions become subtler and more difficult when a model predicts a specific nationality of a name.

To predict the nationality of a name, a model must consider several different factors of a country. First, the language of a country is one of the most important factors to consider. For example, "Chenglong Zhang" is an easily recognizable Chinese name. Second, geographical location can provide useful information for name to country prediction. Germany and Austria are geographically close to each other and categorized as Germanic countries. Therefore, personal names of Germans and Austrians are difficult to separate. Third, the history of a country sometimes plays a crucial role for the naming convention. In Mexico, the Aztec empire in the 16th century was controlled by Spain until 1821. Due to this historical fact, Mexican names tend to look like Spanish names. For example, "Jose Calderson" is a Spanish name and "Jose Corona" is a Mexican name. Other subtle factors can effect the naming conventions varying from country to country. "Von" and "Van" can be one example. The prefix "Von" is related to German origin (e.g., Anake Von Seck, Ida Von Nagel) while "Van" is commonly used in Dutch names (e.g., Wouter Van Pelt, Eake Van Nes). Due to its diversity and subtlety, naming conventions of all countries are quite difficult to distinguish. Table 1 shows how difficult it is to distinguish countries by personal names.

Personal Name	Ethnicity	Nationality
John Stephens	English	Australia
John Boland	English	Great Britain
Reinhold Senn	German	Austria
Wilhelm Baumann	German	Germany
Douglas Rodriguez Guardiola	Spanish	Cuba
Pedro José Gamarro	Spanish	Venezuela

Table 1: Ethnicity and nationality of personal names

On the other hand, as Web services grow, personal names have become one of the most abundant sources of data on the Web [Ambekar *et al.*, 2009; Chang *et al.*, 2010; Treeratpituk and Giles, 2012; Huang *et al.*, 2014; Bergsma *et al.*, 2013]. Results of ethnicity classification have been often utilized as key features of other classification tasks [Wu *et al.*, 2014; Humphreys *et al.*, 2016]. If one can predict or suggest proper countries for personal names, services such as web browsers or mobile applications can benefit from knowing each user's nationalities. However, leveraging personal names for nationality prediction is quite challenging, due to their simple but diverse and subtle structures.

We propose a recurrent neural network based model which predicts nationalities of each name with considerably high accuracy. Without specifying any hand crafted features of naming conventions, our model's performance is comparable with that of previous feature based models. While previous models mainly focused on predicting the ethnicity of a name, we propose to predict the nationality of a person solely based on characters of a name. Character level embeddings dramatically improves the performance, proving its effectiveness in automatic feature extraction. We also evaluate our model on

the ethnicity classification task to test the generalization of our model. To further show the effectiveness of our model, we provide a qualitative analysis on incorrect answers and character embeddings.

Contributions of this paper are three-fold.

- We propose a recurrent neural network based model, predicting the nationality of a personal name solely based on the characters without using external resources or hand crafted features.
- Using character embedding, our model automatically extracts character level features for the name nationality classification task.
- Compared to previous feature based models, our model achieves comparable or better performances on the name nationality classification and name ethnicity classification tasks.

In Section 2, we introduce related studies on personal name based classification tasks and recurrent neural networks. In Section 3, we briefly describe our task and describe our recurrent neural network based model. We provide the Olympic records dataset and the data processing rules in Section 4. In Section 5, we present our results on the nationality and ethnicity classification tasks. In Section 6, we qualitatively show how our model successfully predicts nationalities using automatic feature extraction. Conclusion and future work are presented in Section 7. Our source code and datasets for the experiments are publicly available on the Web. [1]

2 Related Work

2.1 Personal Name Classification

Despite the simplicity of name data, there are a number of studies that utilize personal names for classification tasks.

Ethnicity classification is one of the main tasks that utilizes personal names. [Ambekar *et al.*, 2009] classified 13 cultural groups using decision tree and the Hidden Markov Model on a news corpus. [Treeratpituk and Giles, 2012] classified personal names into 12 ethnic groups based on phonetic sequences information and the character sequences information. [Harris, 2015] extracts race and ethnicity different from classical classify-and-aggregate approaches. Classify-and-aggregate approaches focused on individual classification of names while [Harris, 2015] considered the proportions of each unique name.

There are some following studies that utilize the results of name ethnicity classification task. [Wu *et al.*, 2014] used same model in [Treeratpituk and Giles, 2012] to analyze each ethnicity group's proportion in the computer science research community. [Humphreys *et al.*, 2016] use the ethnicity classification method to find out the relationship between house marketing and ethnicity and prove the cultural superstitions of Chinese.

Name classification models can be also used for classifying users in social networks. [Chang *et al.*, 2010] trains a classifier using a Bayesian approach with U.S. Census name data. The classifier found relationships between Facebook user names and ethnicities. [Pennacchiotti and Popescu, 2011] classifies Twitter users into nationalities or ethnicities using several machine learning 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.*, 2013]. Instead of utilizing user profiles, [Bergsma *et al.*, 2013] used clusters of users' first names, last names, and locations to identify users. [Liu and Ruths, 2013] focuses on gender classification in Twitter using SVM-based classifier and first names. [Huang *et al.*, 2014] utilized Twitter based ethnicity classification results and other information such as user profiles to predict a user's nationality.

[Treeratpituk and Giles, 2012] is one of the studies most similar to this paper. They utilized crawled Wikipedia data for name ethnicity classification task and used multinomial logistic regression for the task. Their model uses various features from basic n-grams to external features such as Double Metaphone or Soundex. Because [Treeratpituk and Giles, 2012] did not share their algorithm or the dataset, we reconstructed the feature based algorithm proposed in their paper and compared the model to our model on the name nationality classification task. As far as we know, this is the first work to classify nationalities based solely on personal names. Also, as [Treeratpituk and Giles, 2012]'s model was built for the ethnicity classification task, we adjusted our dataset for the ethnicity classification task for a fair comparison. We mapped each country to a probable ethnicity using specific rules on each country's race ratio.

2.2 Recurrent Neural Networks

Recurrent neural networks are known for their ability to predict sequential data such as natural language. [Mikolov *et al.*, 2010] showed that recurrent neural networks is effective on language modeling. [Bahdanau *et al.*, 2014] used recurrent neural networks for machine translation, achieving performances comparable with statistical machine translation models.

However, due to its recurrent structure, recurrent neural networks tend to suffer from long-term dependency [Bengio *et al.*, 1994] and severe overfitting problems [Zaremba *et al.*, 2014]. To learn long-term dependencies, [Hochreiter and Schmidhuber, 1997] suggested Long Short Term Memory which significantly reduced the long term dependency problem using memory cell and forget gate. Similar RNN cells such as Gated Recurrent Unit (GRU) [Cho *et al.*, 2014] were introduced to improve efficiency of LSTM. Overfitting problems of RNNs were alleviated with applying dropout on non-recurrent connections of RNNs [Zaremba *et al.*, 2014].

As a result, recurrent neural networks achieve state-of-the-art performance on sequential data especially in natural language processing. Image captioning [Xu *et al.*, 2015], question answering [Kadlec *et al.*, 2016], and machine translation [Bahdanau *et al.*, 2014] all benefited from recurrent neural networks. Also, in most of the previous studies, they utilize an ensemble version of neural networks to boost each model's performance. It is empirically well known that the neural network ensemble works well in many classification tasks [Zhou *et al.*, 2002]. We propose to use recurrent neural networks for

[1] https://github.com/63coldnoodle/ethnicity-tensorflow

name nationality and ethnicity classification tasks. We build a single model that resembles the structures of ensemble neural networks.

2.3 Character Embedding

The concept of dense representation of words was first introduced by [Bengio *et al.*, 2003]. They mapped each word to an embedding lookup table and used resulting dense vectors for language modeling. [Collobert *et al.*, 2011] further improved the idea of word embedding and proved the versatility of word embedding in multiple NLP tasks. After the development of more efficient word embeddings [Mikolov *et al.*, 2013; Pennington *et al.*, 2014], word embedding became a basic component of neural network based natural language processing.

Also, character level embeddings gained popularity among researchers. [Kim *et al.*, 2015] used character embedding to construct word level representations to deal with the "out of vocabulary" problem. [Chiu and Nichols, 2015] also used character embeddings with a convolutional neural network for named entity recognition. However, none of them utilized character embeddings for personal name classification.

3 Task & Model Description

3.1 Name Nationality (Ethnicity) Classification

The objective of the task is to predict the nationality or ethnicity of a personal name. Given a personal name $X = [x_0, x_1, ..., x_{n-1}]$ where x_k indicates kth character of a personal name with maximum length n. Note that x_k can be any n-gram character such as a unigram or bigram. Name nationality classification task is then:

$$p(y_i|X) = D(g(X), \theta) \tag{1}$$

where y_i is ith nationality (ethnicity) label and $D(\cdot)$ represents any kind of discriminative model with trainable parameter θ. $g(\cdot)$ is a function for character level feature extraction.

Note that the range of y differs between nationality and ethnicity with the former usually having more than 10 times the number of classes. For our dataset, we used 127 nationalities and 13 ethnicities, which makes the nationality prediction task much more difficult.

3.2 Model Description

To capture the structures of character sequences, we opt for a recurrent neural network with long short term memory [Hochreiter and Schmidhuber, 1997] as our basic model. For basic RNNs, the hidden state of the neural networks are updated as follows:

$$h_t^l = \sigma(W_{xh} h_t^{l-1} + W_{hh} h_{t-1}^l + b_h) \tag{2}$$

where h_t^l is a hidden state of RNN at time step t in lth layer. Note that $h_t^0 = x_t$ for our task. W_{xh}, W_{hh}, and b_h are parameters to learn, and σ is a basic sigmoid function. The probabilities for each class y_i is given as,

$$p(y_i|X) = softmax([W_{hz} h_t^l + b_z]_i) \tag{3}$$

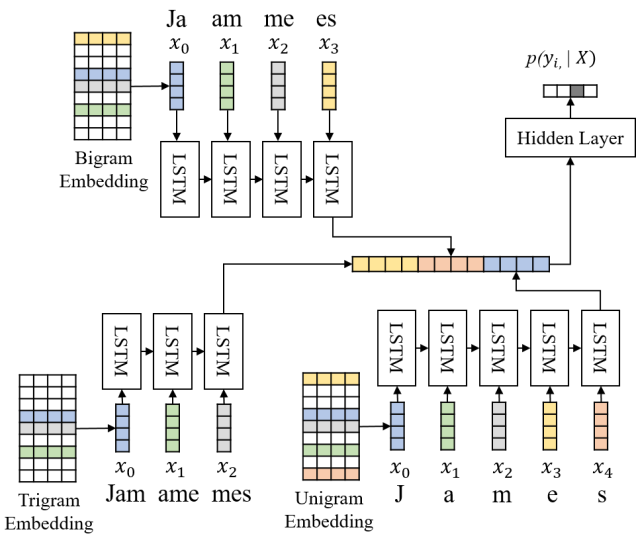

Figure 1: RNN-LSTM model with character embedding

$$softmax(z_i) = \frac{e^{z_i}}{\sum_k e^{z_k}} \tag{4}$$

where W_{hz} and b_z are trainable parameters. However, due to the long term dependency problem, basic RNNs often fail to convey useful information in the hidden states. [Hochreiter and Schmidhuber, 1997] introduced Long Short Term Memory to alleviate the problem using memory cell and forget gate. LSTM is constructed as follows:

$$\begin{aligned}
i &= \sigma(W_{xi} h_t^{l-1} + W_{hi} h_{t-1}^l + b_i) \\
f &= \sigma(W_{xf} h_t^{l-1} + W_{hf} h_{t-1}^l + b_f) \\
o &= \sigma(W_{xo} h_t^{l-1} + W_{ho} h_{t-1}^l + b_o) \\
g &= \tanh(W_{xg} h_t^{l-1} + W_{hg} h_{t-1}^l + b_g) \\
c_t^l &= f \odot c_{t-1}^l + i \odot g \\
h_t^l &= o \odot \tanh(c_t^l)
\end{aligned}$$

where W_* and b_* are trainable parameters. $\odot$ indicates element-wise multiplication. Long term memory can be restored from c_t^l where the information is written with input gate i and previous cell c_{t-1}^l, and updated with forget gate f. After retrieving the hidden state of LSTM, probabilities for each class are the same as Equations 3 and 4. While training the model, LSTM suffered severe overfitting on the training data. To mitigate the problem, we added a dropout to non recurrent part of the RNN as described in [Zaremba *et al.*, 2014].

As inputs x_t are represented as one hot encoding, it is difficult to capture the semantics of each character with respect to the naming convention. Thus, we applied character level embedding to the input x_t to learn more dense representations. From one hot encoded x_t, new input vector x'_t is given as follows:

$$x'_t = P x_t \tag{5}$$

	Raw	Cleaned
# of unique names	17721	17653
# of unique characters	82	42
# of country	127	95
Average length of names	15.3	19.8
Most frequent country	USSR	USSR
Most infrequent country	Guatemala[3]	Bahamas
# of training data	10633	10592
# of validation data	3545	3531
# of testing data	3543	3530

Table 2: Olympic records dataset statistics

	Raw	Cleaned
# of unique names	10449	10308
# of unique characters	82	42
# of ethnicity	13	12
Average length of names	13.2	10.5
Most frequent ethnicity	GER	GER
Most infrequent ethnicity	VIE	AFR
# of training data	6252	6281
# of validation data	2100	2061
# of testing data	2097	2081

Table 3: Olympic records dataset statistics (ethnicity)

where $P \in \mathbb{R}^{d \times K}$ is a projection matrix which turns 1 of K encoded vectors into corresponding dense vectors $x'_t \in \mathbb{R}^d$. We initialized character embeddings in two different ways, random uniform of [-0.1, 0.1] and Skip-gram [Mikolov *et al.*, 2013]. When using Skip-gram, we constructed a training corpus consisting of characters of each name. When using Skip-gram, projection matrix P is obtained by maximizing the following log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq k\leq c, k!=0} \log p(P_{t+k}|P_t) \quad (6)$$

where T is total number of characters of all names, c is the size of context window and P_t is the t-th column vector of P. We used window size of 5 (i.e., c=5) and pretrained the vectors for 10 iterations. Details of Skip-gram can be found in [Mikolov *et al.*, 2013]. In the experiments, we have found that Skip-gram pretrained models achieve better and more robust performances than randomly initialized models. To learn various n-gram features, we constructed three different projection matrices for unigram, bigram, and trigram. Each of them were inputted to three different recurrent neural networks as shown in Figure 1. We concatenate hidden vectors of each RNN and predict the class label y_i. We found that an additional hidden layer after the concatenation gave a slight improvement of performances. Finally, we define and optimize our cross entropy loss function $L(\theta)$ as follows:

$$L(\theta) = -\sum log(p(y_i|X)) \quad (7)$$

We trained our model with back propagation through time [Werbos, 1990].

4 Dataset

We crawled Olympic records data from official Olympic website[2]. Total 17721 pairs of personal names and nationalities were collected and statistics of the dataset is in Table 2.

We processed the data into two different ways. First, we did not preprocessed the raw crawled data, preserving all the capital letters and special characters such as quotation marks or parenthesis (e.g., Anton (Toni) ROM). There exist 11 different non-alphabetic special characters in this dataset, which we call a raw dataset. In this raw dataset, there exists an extreme scarcity of personal names where some countries have only 1 personal name.

For cleaned dataset, we preprocessed the raw dataset as follows: we removed all the redundant characters such as (“) or (-), and lowercased the names. Names between quotation marks or parenthesis were also removed because they were usually used for nicknames. We substituted ‘-’ with spacing. Motivated by [Treeratpituk and Giles, 2012], we added special tokens such as ‘$’ and ‘+’ around the first (middle) names and the last name (e.g. $anton$ +rom+). This resulted in the following five special characters: spacing ( ), apostrophe (’), dollar sign ($), plus sign (+) and a period (.), all of which we considered meaningful for naming convention. Also, scarce country labels that have less than 5 personal names were removed from the cleaned dataset.

For the fair comparisons between models, we made a country ethnicity mapping table for the ethnicity classification task. Since there exist various ethnicities in a country, the majority ethnicity (more than 50%) of a country was considered to be the ethnicity of that country. This gave us total 44 countries to complete ethnicity mappings. For cleaned data, we added additional ethnicity ‘African’ which was not in the previous ethnicity taxonomy. Though the dataset became smaller, the size of the data was large enough to prove the model’s capability.

5 Evaluation

Evaluations are completed on 2 main tasks; Name nationality classification and Name ethnicity classification. Top 1 and top 5 accuracy were used as performance metric. Baseline models and our LSTM based model used for the evaluations are as follows:

- Multinomial Logistic Regression (LR): model suggested by [Treeratpituk and Giles, 2012]
- Random Forest (RF): Random forest model with same feature as LR
- Long Short Term Memory (LSTM): our proposed model

5.1 Model Parameters and Features

Finding the optimal hyper parameters is one of the most important tasks for optimizing neural networks. Using LSTM model, we performed parameter validation on 5 different hyper parameters as described in Table 4. Best performing parameter set on validation phase was used for the LSTM model.

We used Adam optimizer [Kingma and Ba, 2014] for the

[2]http://www.olympic.org

[3]18 countries had only one names each.

Hyper parameter	Range	Step	Optimal
LSTM cell dimension	[50 - 300]	50	200
LSTM dropout	[0.3 - 0.8]	-	0.5
LSTM number of layer	[1 - 3]	1	1
Hidden layer dimension	[100 - 400]	50	200
Learning rate	[1e-4 - 1e-2]	-	0.0035

Table 4: Validation hyper parameters. Steps of each range and selected values are shown.

	Raw	Cleaned
Logistic Regression	44.1 / 74.8	49.3 / 78.2
Random Forest	43.8 / 66.9	46.5 / 68.5
LSTM+uni+bi+tri (embed)	**51.9 / 81.4**	**51.4 / 82.3**

Table 5: Name nationality classification results (top1 / top5 accuracy %)

LSTM model and learning rate decay was set to 0.99 for every 100 iterations. Additionally, mini batch of size 1000 was used. Norms of gradients were clipped at 5. Implementations of RF is based on Sklearn library, and for LR and LSTM, we used Tensorflow [4].

Features for the Logistic Regression and Random Forest were constructed in the same way as [Treeratpituk and Giles, 2012]; nonASCII, character n-gram, Double Metaphone (dmp) n-gram, and Soundex. For character and dmp n-grams, we used bigram, trigram, and four-gram as proposed in [Treeratpituk and Giles, 2012]. Resulting dimension of feature vector is over 66,000 in raw dataset. Note that for our LSTM model, we only utilized character level n-grams (unigram, bigram, and trigram), which are very basic features of personal names. Also, we leveraged unigram features, which were not considered in previous studies for name classification.

5.2 Name Nationality Classification

Results of name nationality classification is in Table 5. We can see that external features of LR and RF have reasonable performances. On both raw and cleaned datasets, LSTM model outperforms RF and LR models with significant margins. Considering the fact that feature based approaches utilized external features such as soundex and double metaphone features, the results are very promising. On cleaned dataset, the LR model's performance improves a lot but accuracy of the LSTM model is still better than LR. We argue that well tuned LSTM models without any external features can effectively predict nationalities of personal names.

To further investigate the effect of each components in our LSTM model, performances of ablated LSTM models on cleaned dataset are reported in Table 6. We constructed one-hot unigram (LSTM+uni), bigram (LSTM+bi), and trigram (LSTM+tri) based LSTM model. Embedded versions for each n-gram are marked with (embed). Despite the lack of other n-gram features, the performances of ablated models are considerably high compared to the Random Forest model. We can see that bigram and trigram embeddings have significantly boosted the performance. Unigram embeddings, which were not considered in feature based approaches, also shows its effectiveness when used alone.

[4] http://www.tensorflow.org

	Accuracy
LSTM+uni	47.8 / 81.2
LSTM+uni (embed)	48.6 / 80.1
LSTM+bi	52.1 / 82.5
LSTM+bi (embed)	52.2 / 82.0
LSTM+tri	52.1 / 80.1
LSTM+tri (embed)	52.6 / 82.4

Table 6: Name nationality classification LSTM model ablation (top1 / top5 accuracy %)

	Raw	Cleaned
Logistic Regression	78.3 / 96.7	**82.5** / 98.1
Random Forest	73.9 / 92.8	75.5 / 94.0
LSTM+uni+bi+tri (embed)	**84.8 / 98.0**	81.7 / **98.7**

Table 7: Name ethnicity classification results (top1 / top5 accuracy %)

5.3 Name Ethnicity Classification

Results of ethnicity classification is in Table 7. Using the ethnicity mapping table, most of the models accurately identify the ethnicity of each name. The performance of Logistic Regression is similar to that of [Treeratpituk and Giles, 2012] (85%), suggesting that the mapping was reasonable.

Among all the models, LSTM model performed similarly to the LR model. Though LR model's performances have slightly higher top 1 accuracies in cleaned dataset, LSTM showed better top 5 accuracies once again. It seems that the LSTM model captures broader concepts of naming conventions. In our detailed analysis, KOR and JAP ethnicities were the easiest to discriminate on both LR and LSTM models.

6 Qualitative Analysis

By inspecting incorrect predictions of our model, we present qualitative analysis of the models. In Table 8, we give lists of the top 5 predictors for incorrect predictions.

From the Table, we can see that the incorrect answers of LSTM are also reasonable. When predicting nationality of "$yoanka$ +gonzalez+", both LR and LSTM correctly predict "Cuba". Though RF model identified the correct answers in the top 5, the listed countries seemed rather unrelated (Japan, Germany). On the other hand, LSTM model suggests Spanish using countries such as Mexico or Urguay. "$oscar$ +brayson+" is a Cuban name which has Spanish naming custom, although it has quite globally common first name "Oscar". Our model captures the features of both "Oscar" and "Brayson", and suggests proper nationalities. When predicting the nationality of "$georg$ +tallberg+", RF and LR focused on German custom "-berg", but failed to suggest Finland. Based on the fact that "Georg" is a common northern European name, LSTM model successfully suggests Finland, as well.

To analyze how our LSTM based models automatically extracted features from characters, we plotted each character embedding into lower dimensional space using T-SNE

	Random Forest	Logistic Regression	LSTM+uni+bi+tri (embed)
$yoanka$ +gonzalez+	Netherlands Japan Hungary Germany **Cuba**	**Cuba** Bulgaria Mexico Germany Argentina	**Cuba** Mexico Uruguay Canada Italy
$oscar$ +brayson+	Germany Great Britain Australia Uruguay Portugal	Australia Belgium United States Of America Canada Germany	France United States of America Australia Norway **Cuba**
$georg$ +tallberg+	Federal Republic of Germany Sweden Norway Switzerland Denmark	Sweden Germany United Team of Germany New Zealand Denmark	Sweden Norway Azerbaijan Germany **Finland**

Table 8: Top 5 answers of RF, LR and LSTM model

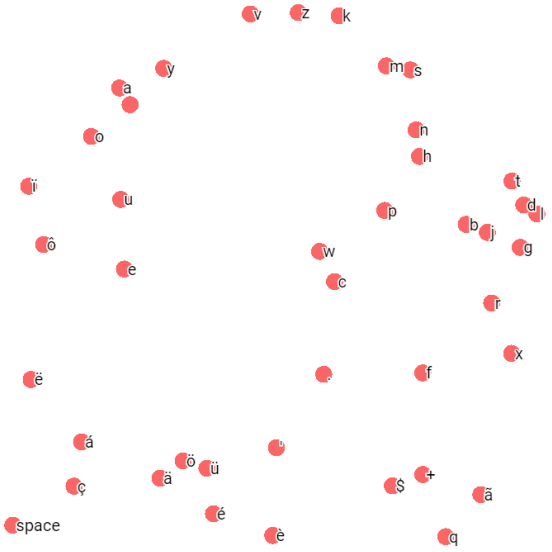

Figure 2: T-SNE Visualization results of Unigram Embeddings (Cleaned dataset)

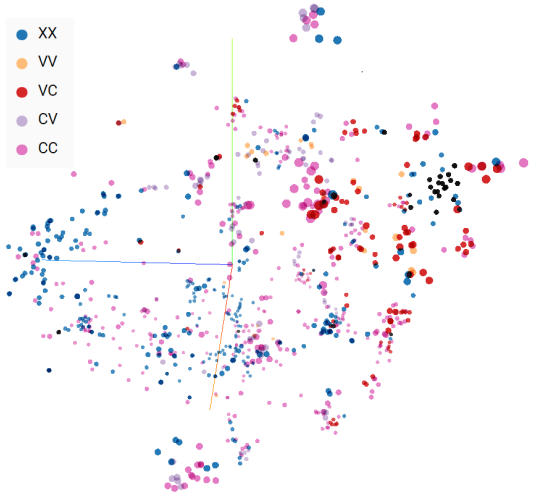

Figure 3: T-SNE Visualization results of Bigram Embeddings (Cleaned dataset)

[Maaten and Hinton, 2008] in Figure 2 and 3. In unigram embeddings, we can clearly see that the model separates vowels, consonants, European characters, and special characters. For instance, the nearest neighbors of 'a' are 'e', 'i', 'o', 'u'. Special characters that mark the boundaries ('$' and '+') were also mapped closely. In bigram embeddings, we plotted vowel+vowel (VV, yellow), vowel+consonant (VC, red), consonant+vowel (CV, light pink), consonant+consonant (CC, pink), and others (XX, blue). Bigram embeddings are plotted more complex than unigram embeddings, but the characters tend to make clusters based on their combinations of vowels and consonants. The nearest neighbors of 'bo' are 'to', 'ko', and 'go' in bigram embeddings. As a result, we can see that n-gram embeddings have succeeded in automatically extracting character level features for name classification tasks.

7 Conclusion

Predicting nationality based solely on personal names has been poorly studied due to its difficulties finding subtle structures between naming conventions. We propose a recurrent neural network based model, which predicts nationality of each personal name with high accuracy. Utilizing Skip-gram based embeddings, we show that n-gram embeddings significantly improve the performance our models.

For future work, we can produce various n-gram embeddings utilizing only unigram embeddings. Also, hierarchical recurrent neural networks can be applied to extract higher and more complex representation of personal names.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the government of Korea (MSIP) (NRF-2014R1A2A1A10051238).

References

[Ambekar *et al.*, 2009] 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.

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

[Bengio *et al.*, 1994] 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.

[Bengio *et al.*, 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[Bergsma *et al.*, 2013] Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, and David Yarowsky. Broadly improving user classification via communication-based name and location clustering on twitter. In *HLT-NAACL*, pages 1010–1019, 2013.

[Chang *et al.*, 2010] Jonathan Chang, Itamar Rosenn, Lars Backstrom, and Cameron Marlow. epluribus: Ethnicity on social networks. *ICWSM*, 10:18–25, 2010.

[Chiu and Nichols, 2015] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[Collobert *et al.*, 2011] 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.

[Harris, 2015] J Andrew Harris. WhatâĂŹs in a name? a method for extracting information about ethnicity from names. *Political Analysis*, 23(2):212–224, 2015.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Huang *et al.*, 2014] Wenyi 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.

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

[Kadlec *et al.*, 2016] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*, 2016.

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

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

[Liu and Ruths, 2013] 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.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

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

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Pennacchiotti and Popescu, 2011] Marco Pennacchiotti and Ana-Maria Popescu. A machine learning approach to twitter user classification. *Icwsm*, 11(1):281–288, 2011.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[Treeratpituk and Giles, 2012] Pucktada Treeratpituk and C Lee Giles. Name-ethnicity classification and ethnicity-sensitive name matching. In *AAAI*, 2012.

[Werbos, 1990] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[Wu *et al.*, 2014] Zhaohui Wu, Dayu Yuan, Pucktada Treeratpituk, and C Lee Giles. Science and ethnicity: How ethnicities shape the evolution of computer science research community. *arXiv preprint arXiv:1411.1129*, 2014.

[Xu *et al.*, 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81, 2015.

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

[Zhou *et al.*, 2002] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.