VIETNAM NATIONAL UNIVERSITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER ENGINEERING

Natural Language Processing Report

Vietnamese name correction using LSTM Language Model

Student: Nguyễn Ngọc Hải Đăng
MSHV: 1970513

HO CHI MINH CITY, 2020

Mục lục

1 Dataset

1.1 Description

The dataset is a .csv file containing the Vietnamese full name of both male and female, the name structure follows the conventional structure in Vietnam, for example: ”Nguyen Ngoc Hai Dang". The dataset has 8758 records, and a total size of 221KB.

The source of this name collection is from this Github link https://github.com/duyet/vietnamese-namedb-crawler. However, the original is a .json file with more detail for each record, so I extracted from it only the ”Full name" attribute for use in this assignment and put it in the mentioned .csv above.

1.2 Data Preprcessing

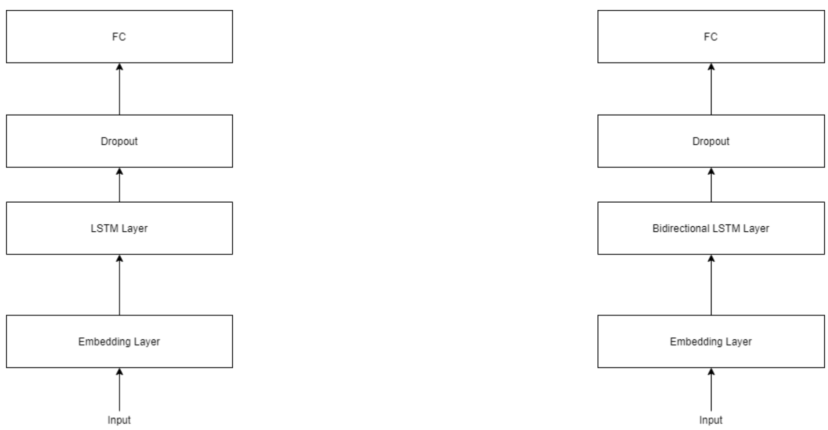
The dataset is preprocessed in the following order:

- The dataset is augmented by duplicating each record 20 times.
- Add characters that will be used as the beginning of sequence and end of sequence. { will be the beginning of sequence and } will be the end.
- Concatenate the dataset then truncate it into fixed-length sequences. The sequence length used in the below experiments is 15.
- Each character sequence is encoded into an integer sequence using the dictionary generated from the dataset. This sequence will then be encoded into one-hot sequences to be used as input for the training process

2 Methodology

2.1 Network Architect

In this assignment, I used two different models to do the spelling correction on character-level. One is a conventional LSTM and the other is the Bi-directional LSTM as shown in the figure below. The two networks use the same architect with the only difference is the LSTM Layer and the Bi-directional LSTM Layer. The idea behind bi-directional LSTM involves duplicating the first recurrent layer in the network so that there are now two layers side-by-side, then providing the input sequence as-is as input to the first layer and providing a reversed copy of the input sequence to the second.



(a) Standard LSTM (b) Bi-directional LSTM

Hình 1: LSTM Networks

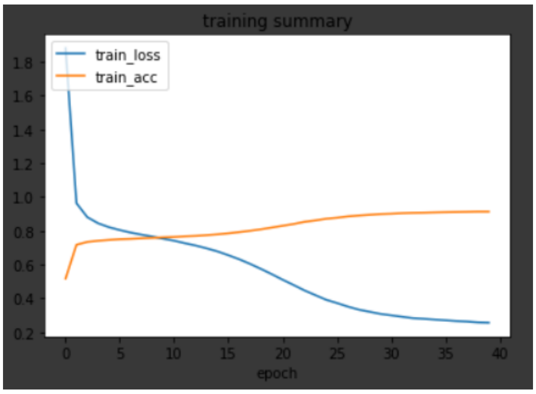
The input will be a matrix that represents a sequence of characters, which then go through an embedding layer for encoding. The output of the LSTM layer will go into the dropout layer which serves as a regularization layer, and the final fully connected layer uses sigmoid function as activation to transform the result into a distribution interpreted as the probability of the next character after the input sequence.

2.2 Training process

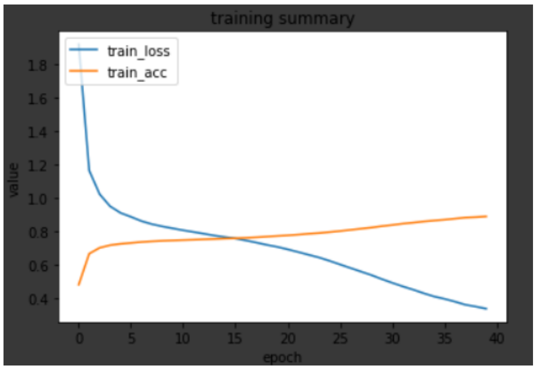
The training process for each model use the same configuration:

- Batch size is 256.
- Number of epochs is 40.
- Each input is a one-hot sequence of 15 characters.
- The model is saved for every 5 epochs.

The training process of each model is shown in the following figure.



(a) Standard LSTM training process



(b) Bi-directional LSTM training process

Hình 2: Training process

Due to the addtional layer in bidirectional LSTM, the layer that receives inverse input, the time required for training Bi-LSTM is significantly higher than standard LSTM and the its loss also decreases slower comparing to the standard one.

3 Using the Language Model

3.1 Predicting the next character

In this use case, we will use the trained models for predicting the next possible characters of a sequence with the output will show the top 5 most possible characters for a sequence. The input is handled just like each of the sequences used for training.

The predictions of for each sequence when using the two model is shown in the below figures.

```
Input: Nguyễn Ngọc H
Output 1 : Nguyễn Ngọc Ho       Probability: 0.27084655
Output 2 : Nguyễn Ngọc Hi       Probability: 0.24741389
Output 3 : Nguyễn Ngọc Hu       Probability: 0.15211129
Output 4 : Nguyễn Ngọc Hả       Probability: 0.11944721
Output 5 : Nguyễn Ngọc Hư       Probability: 0.088363044


Input: Nguyễn Ngọ
Output 1 : Nguyễn Ngọc   Probability: 0.9999608
Output 2 : Nguyễn Ngọa   Probability: 2.7465056e-05
Output 3 : Nguyễn Ngọg   Probability: 3.7544305e-06
Output 4 : Nguyễn Ngọ    Probability: 3.299384e-06
Output 5 : Nguyễn Ngọọ   Probability: 1.6404052e-06


Input: Nguyễn Tha
Output 1 : Nguyễn Than   Probability: 0.99997234
Output 2 : Nguyễn Thao   Probability: 2.6678987e-05
Output 3 : Nguyễn Tha}   Probability: 3.9194182e-07
Output 4 : Nguyễn Tha    Probability: 3.4273978e-07
Output 5 : Nguyễn Thaé   Probability: 1.7897695e-07
```

(a) LSTM model result

```
Input: Nguyễn Ngọc H
Output 1 : Nguyễn Ngọc Hả       Probability: 0.34375045
Output 2 : Nguyễn Ngọc Ho       Probability: 0.2058964
Output 3 : Nguyễn Ngọc Hu       Probability: 0.18701218
Output 4 : Nguyễn Ngọc Hư       Probability: 0.11237474
Output 5 : Nguyễn Ngọc Hi       Probability: 0.050249156


Input: Nguyễn Ngọ
Output 1 : Nguyễn Ngọc   Probability: 0.9999814
Output 2 : Nguyễn Ngọt   Probability: 7.642575e-06
Output 3 : Nguyễn Ngọ    Probability: 6.728513e-06
Output 4 : Nguyễn Ngọ}   Probability: 3.7497455e-06
Output 5 : Nguyễn Ngọy   Probability: 2.2228087e-07


Input: Nguyễn Tha
Output 1 : Nguyễn Than   Probability: 0.99910897
Output 2 : Nguyễn Tha    Probability: 0.0005168207
Output 3 : Nguyễn Thao   Probability: 0.00016705426
Output 4 : Nguyễn Tha}   Probability: 4.324811e-05
Output 5 : Nguyễn Thaá   Probability: 3.1218282e-05
```

(b) Bi-LSTM model result

Hình 3: Predicting the next word possible words

3.2 Correcting mistakes

We will try to detect and correct the error that is near the end of sequence in this experiment with an assumption that the first 8 characters are correct. These 8 characters will cover a person's last name and some characters of the middle names.

After feeding to model with the 8-character sequence, we will start predicting from the 9th character of the input sequence, and if the actual character in the respective position of input has a lower probability than a threshold, we will replace the character with one that has a higher probability.

```
Input: Nguyễn Ngic
output 1: {Nguyễn Ngọc -- with probality: 0.5132400989532471
output 2: {Nguyễn Nguc -- with probality: 0.4857058525085449
output 3: {Nguyễn Ngôc -- with probality: 0.000614945194683969
output 4: {Nguyễn Ngoc -- with probality: 0.00037276404327712953
output 5: {Nguyễn Nghc -- with probality: 1.5370249457191676e-05
Input: Nguyễn Tuen
output 1: {Nguyễn Tuấn -- with probality: 0.9985476136207581
output 2: {Nguyễn Tuyn -- with probality: 0.001383060938678681 9
output 3: {Nguyễn Tuản -- with probality: 1.4468570043391082e-05
output 4: {Nguyễn Tuần -- with probality: 1.3854332792107016e-05
output 5: {Nguyễn Tuến -- with probality: 9.408693586010486e-06
Input: Nguyễn Thanh Sương
output 1: {Nguyễn Thanh Sươn -- with probality: 0.9999983310699463
output 2: {Nguyễn Thanh Sươ} -- with probality: 1.5728863900221768e-06
output 3: {Nguyễn Thanh Sươm -- with probality: 1.6514681533408293e-07
output 4: {Nguyễn Thanh Sương -- with probality: 1.3414212851614593e-08
output 5: {Nguyễn Thanh Sươ  -- with probality: 5.587589946287608e-09
```

(a) LSTM model result

```
Input: Nguyễn Ngic
output 1: {Nguyễn Ngọc -- with probality: 0.9889529943466187
output 2: {Nguyễn Ngậc -- with probality: 0.00727052753791213
output 3: {Nguyễn Ngôc -- with probality: 0.0005654986016452312
output 4: {Nguyễn Nghc -- with probality: 0.0005465794238261878
output 5: {Nguyễn Ngấc -- with probality: 0.0005340368370525539
Input: Trương Tuen
output 1: {Trương Tuấn -- with probality: 0.9687374830245972
output 2: {Trương Tuan -- with probality: 0.013828330673277378
output 3: {Trương Tuyn -- with probality: 0.009501294232904911
output 4: {Trương Tuân -- with probality: 0.006889774929732084
output 5: {Trương Tu n -- with probality: 0.0003081375616602599 6
Input: Nguyễn Thanh Sương
output 1: {Nguyễn Thanh Sươn -- with probality: 0.9996028542518616
output 2: {Nguyễn Thanh Sươ} -- with probality: 0.0003876486734952777 6
output 3: {Nguyễn Thanh Sươc -- with probality: 2.4943219614215195e-06
output 4: {Nguyễn Thanh Sườ -- with probality: 2.443226321702241e-06
output 5: {Nguyễn Thanh Sươu -- with probality: 1.9615515611803858e-06
```

(b) Bi-LSTM model result

Hình 4: Correcting one near end error

In addition to the experiment above, we will try to correct the entire input sequence with the same assumptions above. The only difference is that we use the best possible output in each correction step to change the orginal sequence and keep correcting until we reach the intput length.

```
Input: Nguyễn Ngic hai Đnng
mid-output: {Nguyễn Vgic Hai Đnng
mid-output: {Nguyễn Vũic Hai Đnng
mid-output: {Nguyễn Vũ}c Hai Đnng
mid-output: {Nguyễn Vũ} Hai Đnng
mid-output: {Nguyễn Vũ} {ai Đnng
mid-output: {Nguyễn Vũ} {Ni Đnng
mid-output: {Nguyễn Vũ} {Ng Đnng
mid-output: {Nguyễn Vũ} {Ngu Đnng
mid-output: {Nguyễn Vũ} {Nguy Đnng
mid-output: {Nguyễn Vũ} {Nguyễ Đnng
mid-output: {Nguyễn Vũ} {Nguyễn Đnng
mid-output: {Nguyễn Vũ} {Nguyễn Đứng
mid-output: {Nguyễn Vũ} {Nguyễn Đứcg
mid-output: {Nguyễn Vũ} {Nguyễn Đức
Final output: Nguyễn Vũ Nguyên Đức


Input: Trương Tuen Aanh
mid-output: {Trương Euen Aanh
mid-output: {Trương Eu}n Aanh
mid-output: {Trương Eu} Aanh
mid-output: {Trương Eu} {anh
mid-output: {Trương Eu} {Nnh
mid-output: {Trương Eu} {Ngh
mid-output: {Trương Eu} {Ngu
Final output: Trương Eu Ngu


Input: Nguyễn Thenh Suung
mid-output: {Nguyễn Thánh Suung
mid-output: {Nguyễn Thánh Saung
mid-output: {Nguyễn Thánh Sanng
mid-output: {Nguyễn Thánh Sangg
mid-output: {Nguyễn Thánh Sang}
Final output: Nguyễn Thánh Sang
```

(a) LSTM model result

```
Input: Nguyễn Ngic hai Đnng
mid-output: {Nguyễn Ngọc Hai Đnng
mid-output: {Nguyễn Ngọc Hai Đăng
Final output: Nguyễn Ngọc Hai Đăng


Input: Trương Tuen Aanh
mid-output: {Trương Tuấn Aanh
mid-output: {Trương Tuấn Annh
mid-output: {Trương Tuấn Anhh
Final output: Trương Tuấn Anh


Input: Nguyễn Thenh Suung
mid-output: {Nguyễn Thành Suung
mid-output: {Nguyễn Thành Saung
mid-output: {Nguyễn Thành Sanng
mid-output: {Nguyễn Thành Sangg
Final output: Nguyễn Thành Sang
```

(b) Bi-LSTM model result

Hình 5: Correcting the entire sequence

4 Conclusion

The experiments done in 3 show that LSTM can perform well when errors occur near the end of the sequence, but its performance drop significantly for errors near the beginning. On the other hand, the performance of Bi-LSTM is considered stable in all of the experiement.

As shown in 2.2, the performance of both model can increase with further fine-tuning and more data. Furthermore, techniques such as look ahead can be investigated to increase the performance of these models