International Journal of
Computer & Software Engineering
Original Article Open Access

A Novel Method on Translating People's Names in Mandarin - 'LT-NTM'

Hua Zhao* **and Fairouz Kamareddine**
School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK

Abstract

Romanized names are widely used for global communication and interaction. In text translation, translating the romanized names to their origin language names is a challenge. For example, in Mandarin, one Pinyin word has over 100 Hanzi characters. Therefore, translating romanized names to their origin language names is also a big challenge for Human translators. In our study, we found that most of the existing name translation methods only work with European languages. This paper focuses on Mandarin name translation. We propose a novel method for translating Pinyin names to Hanzi characters, which is called 'LT-NTM'. It uses phonetic patterns on word identification. The evaluation shows that our novel method has higher accuracy than Google translate on Mandarin name translation.

Publication History:

Received: December 16, 2022
Accepted: January 07, 2023
Published: January 09, 2023

Keywords:

Machine Learning, Text Translation, Mandarin Name

Introduction

People write their names in roman instead of their first language (e.g., Chinese, Japanese, Arabic, etc.) for global communication and interaction. Here, we define 'Text diversity[1] of people's names' as a person's name in different characters. Translating proper names[2] is a way to analyse people's names for social research. For international business analysis, researchers can translate the proper names into different languages for analysing a business in different countries. However, translating proper names from one language to another is a challenge [2]. In our study, in Mandarin, the romanisation of names is not unique: a Pinyin[3] word can have many different Hanzi[4] versions. For example, a Pinyin word corresponds with over 100 Hanzi characters. In our study, we also found several existing name matching methods that work for translating people's names [4]. However, most of these existing methods only work with European languages, especially in English [5].

In this paper, we propose a novel model to translate Pinyin names to Hanzi characters. And, we focus on the name translation in Mandarin[5]. This novel model is called 'LT-NTM' (Lexical Tone - Name Translation Model). 'LT-NTM' uses regular phonetic patterns to analyse the words relations of a Pinyin name to help translate this Pinyin name to Hanzi characters. 'LTNTM' has two steps to translate a Pinyin name to Hanzi versions. We first generate the Hanzi word of the Pinyin Surname of a Pinyin name. Next, we use the regular phonetic patterns to translate the Pinyin given name to its Hanzi version. We evaluate 'LT-NTM' using the 'Chinese Names Corp' and the '9800 Chinese Names with Gender' data sets (Conclusion). We compared our novel model 'LT-NTM' with 'Google translate' in the evaluation. The experiment results show that 'LTNTM' achieve the function of translating Pinyin names to Hanzi versions (Experiment results). Furthermore, it has better accuracy than 'Google translate' on Mandarin name translation (Experiment results).

Our main contributions are as follows:

1. Building a novel model of text diversity translation in Mandarin names (Data for evaluation).
2. A novel method of four-word name identification in Mandarin names (Experimental setup).
3. A novel method of identifying the labelled Hanzi names in different lengths (Experiment results).
4. Good performance on text diversity translation in Mandarin names (Conclusion).

The organization of this paper is as follows: In section *Challenge*, we indicate the challenge of text diversity translation and the setup of our proposal. Section *'LT-NTM' Overview* introduces the methods in 'LT-NTM'. In section 4, we describe the data setup and model training for the evaluation of the proposed model. In *Experimental setup*, we explain the experimental setup of 'LT-NTM'. In section *Experiment Results*, we report the evaluation results of the proposed model. Section *Conclusion* concludes our work in this article. The preliminary versions of this work appeared in Zhao et al [7].

Challenge

In 2004, Anthony Pym [8] said that proper names could not be translated because of 'SL' (Source language) name and 'TL' (Target Language) name inequivalence. The translation is defined as transferring the written or spoken 'SL' texts to equivalent written or spoken 'TL' texts [9]. After three years, Heikki Särkkä listed four strategies for proper names translation [10]. In 2021, Veronica Sand [11] reported five translation strategies for translating proper names. They are 'kept', 'Direct translations', 'Modified', 'Partly translated' and 'completely altered'.

In text diversity translation, Stephen et al. [12] proposed an algorithm for automatic English-Chinese place name transliteration. Newmark [13] designed a name translator for many languages. However, in our study, we found that there is no existing name translator to translate Pinyin names to Hanzi names. Therefore, we propose a model to translate Pinyin names to Hanzi characters in this paper.

[1]Text diversity generally means that an idea has different world views expressions from different books and genres [1].

[2]Proper name is a word which is the name of a person, a place, an institution, etc. It is written with a capital letter [2, 3].

[3]Pinyin is the translation into the Roman Alphabet of lexical tone transcriptions from Chinese characters.

[4]Hanzi is the Chinese Character.

[5]An official language of Chinese which is spoken by over 730 million people [6].

***Corresponding Author:** Dr. Hua Zhao, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom, E-mail: dr.hua.zhao@gmail.com

Citation: Zhao H, Kamareddine F (2023) A Novel Method on Translating People's Names in Mandarin - 'LT-NTM'. Int J Comput Softw Eng 8: 183. doi: https://doi.org/10.15344/2456-4451/2023/183

In our study, we found some existing online computer translators, such as Google Translate and Bing Translator. These translators have made significant contributions to languages transformation. To help evaluate our proposed models of text diversity translation of people's names in Mandarin, we want to compare them with existing translators. Therefore, we select 'Google translate'.

'LT-NTM' Overview

In this section, we introduce our proposed model 'LTNTM' (Lexical Tone - Name Translation Model). 'LTNTM' uses regular phonetic patterns to analyse thewords relations of a Pinyin name to help translate this Pinyin name to be a Hanzi name.

Regular phonetic patterns and lexical tones

Some references said that Pinyin practice could help to learn Hanzi characters [14,15]. That is because Pinyin illustrates the exact sound of a given Hanzi character [14]. In Mandarin, we use lexical tones to display the pronunciation of a Pinyin word. A Pinyin word can be pronounced in four different lexical tones [16]. These four lexical tones are the high level tone, the low rising tone, the falling-rising tone and the high falling tone [14,16-18]. And, they are the important components of a phonetic word in Mandarin [14]. In our study, we found that Mandarin names have regular phonetic patterns. The pronunciation of Mandarin names is typically mouthful. For example, 'Wang Qing' is a Pinyin name where 'Wang' is the surname and 'Qing' is the Given name (Table 1).

Pinyin	Pinyin with Lexical Tone
Wang	wāng; wǎng
Qing	qīng;qíng;qǐng;qìng

Table 1: Lexical Tones Of Example Name 'Wang Qing'.

In table 1, 'wāng qìng', 'wǎng qīng', 'wáng qíng', 'wáng qìng' are the regular phonetic patterns of 'Wang Qing'. And, in the study of the regular phonetic patterns of 'Wang Qing', we found that no people use 'wàng qìng', 'wāng qīng', 'wǎng qīng' as their phonetic names in 1.2 billion Mandarin names. Therefore, we use the logic of regular phonetic patterns to detect Pinyin names and translate Pinyin names to Hanzi names for Mandarin name translation.

Lexical tones setup for 'LT-NTM'

In 'LT-NTM', we use regular phonetic patterns to translate Mandarin names. Lexical tones are the important components of Pinyin pronunciation [14]. In 'LT-NTM', we use numbers as the lexical tone of Pinyin names.

Lexical Tone	In Mandarin	In LT-NTM
High-Level Tone	wāng	wang1
Low Rising Tone	wáng	wang2
Falling-Rising Tone	wǎng	wang3
High Falling Tone	wàng	wang4

Table 2: Examples Of Lexical Tones In Mandarin And 'LT-NTM'

Table 2 displays the four lexical tones of an example Pinyin word 'Wang'. In 'LT-NTM', we set 'wang1' to display 'wāng' as the high-level tone of Pinyin 'Wang'. We replace 'wáng' to 'wang2' as the low rising tone of Pinyin 'Wang'. We use 'wang3' to display 'wǎng' as the falling-rising tone of Pinyin 'Wang'. And, we replace 'wàng' to 'wang4' as the high falling tone of Pinyin 'Wang' in 'LT-NTM'.

Name presentation setup

To assist our model to understand the structure of these names, we set the presentation style of each name. In a Mandarin name, we add ', between each labelled Surname and labelled Given name. We use '-' to separate each word when the labelled Surname or Given name has more than one word. For example, the example pinyin name 'Yuan Tian Gang' is displayed as 'Yuan, Tian-Gang' in the experiments.

The introduction of 'LT-NTM'

Definitions in 'LT-NTM'

- Pinyin Name 'N': We set a Pinyin name as 'N' in 'LT-NTM'. A Pinyin name in 'LT-NTM' is defined as follows:

$$N = (g_{1,} g_2^l) \qquad (1)$$

where g_1 is the Pinyin Surname of N, g_2^l is a list of Pinyin words of the Given name of N. In a list of Pinyin Given name g_2^l, we set l to be the number of each indexed Pinyin word, where $l \in \{1,2,3\}$.

- Phonetic Pinyin Word: we use 'Phonetic Pinyin Word' to refer to the phonetic of Pinyin words of g_1 and a list of g_2^l. In section *regular phonetic patterns and lexical tones*, we explained the setup of lexical tones in 'LT-NTM'.
- Filter Feature: we use 'Filter Feature' to narrow the quantity of possible Hanzi words of Pinyin Surname 'g_1' and a list of Pinyin Given name 'g_2^l'. In 'LT-NTM', we set four filter features. They are 'Sur_N'(Surname), 'Given_N'(Given Name), 'Gen'(Gender) and 'LT'(Lexical Tone). We first use the filter features of 'Sur_N' and 'Given_N' to narrow the quantity of the possible Hanzi words of Pinyin Surname 'g_1'. We then use the filter features of 'Sur_N', 'Given_N', 'Gen' and 'LT' to narrow the quantity of the possible phonetic Pinyin words and possible Hanzi words of a list of Pinyin Given name g_2^l.

The process of 'LT-NTM'

Figure 1 shows the architecture of 'LT-NTM'. In the figure, LT-NTM translates the Pinyin name 'N' which is set as 'g_1 , $g_{2^1} - g_{2^2}$' to Hanzi name 'C_2, $F_{1,4} - F_{2,3}$'. Our proposed 'LT-NTM' has two steps to translate Pinyin names to Hanzi names. We call the first step 'Surname Translation', the second step is 'Regular Phonetic patterns and Given Name Translation'.

Step 1: In the first step, we generate the Hanzi characters of Pinyin surname 'g_1'. We first generate a list of Hanzi words 'C_i' (where $i \in \{0,1,2, .., q\}$) from the Pinyin surname g_1. In figure 1, C_1 and C_2 are the identified Hanzi words of Pinyin Surname g_1. We then use the filter features of 'Sur_N' and 'Given_N' to verify the list of Hanzi words C_i. In figure 1, 'C_2' is the translated Hanzi character of Pinyin surname 'g_1'. At the end of this step, we convert the translated Hanzi character 'C_2' to a Pinyin word with Lexical tone, 'S_2'.

Step 2: In the second step, we translate a list of Pinyin Given name 'g_2^l' to Hanzi characters. To translate the Given name of a Pinyin name

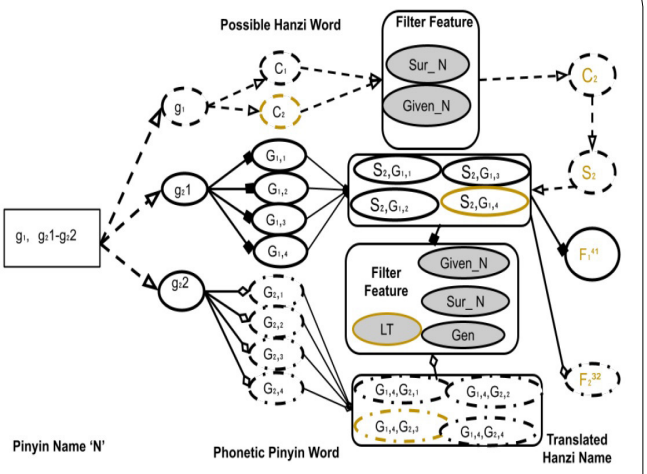

Figure 1: The Architecture of 'LT-NTM'.

to Hanzi name, we apply regular phonetic patterns. In step two, we first assign four lexical tones of a list of Pinyin Given name g_2^i. We then pair the last translated phonetic word with a list of generated phonetic words of a proper Pinyin Given name. Here, we use the filter feature of 'LT' to narrow the quantity of the paired phonetic names. In the end, we use the filter features of 'Given_n', 'Sur_N' and 'Gen' to verify the Hanzi word of phonetic Pinyin Given name.

Methods in 'LT-NTM'

In the last section, we described that 'LT-NTM' has two steps to translate a Pinyin name to Hanzi characters. They are 'Surname Translation' and 'Regular Phonetic patterns and Given Name Translation'.

Step one - surname translation

In the step of 'Surname Translation' in 'LT-NTM', we aim to translate the Pinyin Surname of a Pinyin name to Hanzi characters.

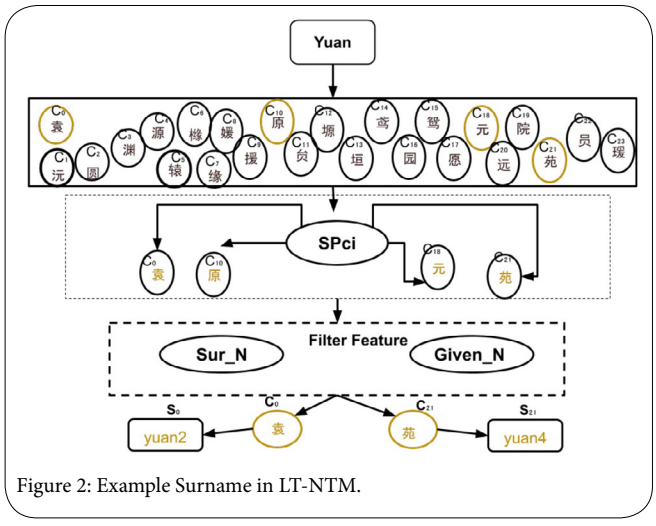

Figure 2: Example Surname in LT-NTM.

We first generate a list of possible Hanzi words of the Pinyin Surname. We then use 'StartPro' to narrow the quantity of the list of generated possible Hanzi words, which is applied from the method of 'DT-NTM'. In the end, we use the filter features of 'Sur_N' and 'Given_N' to narrow and verify the quantity of the list of generated possible Hanzi words (Figure 1). Here, we use the method of 'JudgProS' to filter the possible Hanzi words. We will explain these methods by using an example Pinyin name 'Yuan, Tian-gang'.

Figure 2 shows the process of translating Pinyin Surname 'Yuan' of the example Pinyin name 'Yuan, Tiangang' by using 'LT-NTM'.

In figure 2, we first generate a list of possible Hanzi words of Pinyin Surname 'Yuan'. Here, 'LT-NTM' generates twenty-four possible Hanzi words for Pinyin Surname 'Yuan'. We generate these possible Hanzi words using the training data set 'Learning for LTNTM' (Experimental setup). We then use the method of 'StartPro' to narrow the generated 24 possible Hanzi words of Pinyin surname 'Yuan'. In 'LT-NTM', we set the result of 'StartPro' as ' SP_{Ci} ', which is defined as follows:

$$SP_{Ci} = \frac{C_i(s)}{\sum_{i=1}^{i} C_i(s)} \quad (2)$$

$$SP_{Ci} > 0.01 \Rightarrow C_i \quad (3)$$

Here, C_i is a list of generated possible Hanzi words of Pinyin Surname, where $i \in \{0,1,2, ..,q\}$. And, $C_i(s)$ is the frequency of the Hanzi word C_i as a surname (see Section 5.1). We set the condition of the result ' SP_{Ci}^{i} ' that should be larger than 0.01. In figure 2, the results of narrowing the possible Hanzi words of the Pinyin Surname 'Yuan' are '袁', '原', '元' and '苑' by using 'StartPro' (Equation 2).

Next, we use the filter features of 'Sur_N' and 'Given_N' to verify the generated possible Hanzi words. Here, we use the method of 'JudgProS', which is shown below,

$$GP_{C_i} = \frac{c_i(sVg)}{c_i(s) + c_i(g)} \quad (4)$$

$$SP_{C_i} > GP_{C_i} \Rightarrow C_i \quad (5)$$

Here, we set the result of the method 'JudgProS' as 'GP_{C_i}'. And, $C_i(s)$ is the frequency of the Hanzi word C_i as a surname (Experimental setup). $C_i(g)$ is the frequency of the Hanzi word C_i as a Given name (Experimental setup). We set the condition of the result 'SPCi' that should be larger than 'GP_{C_i}'. In figure 2, '袁' and '苑' are the translated Hanzi Surname of Pinyin 'Yuan' by using the method of 'JudgProS'.

At the end of this step, we generate the phonetic Pinyin words for the translated Pinyin Surname. In figure 2, 'yuan2' is the phonetic Pinyin word of '袁' and 'yuan4' is the phonetic Pinyin word of '苑'.

Step Two - Regular Phonetic patterns and Given Name Translation

In the 'STEP 2' of 'LT-NTM', we aim to translate the Pinyin Given name of a Pinyin name to Hanzi characters. We first generate the phonetic Pinyin words of the Pinyin Given name.

We then pair the phonetic Pinyin word of the last translated proper name with the generated phonetic Pinyin word of the Pinyin Given name. Then, we use the filter feature of 'LT' to narrow the quantity of the paired phonetic names. Next, we translate the selected phonetic Pinyin Given name to Hanzi characters. In the end, we use the filter features of 'Sur_N', 'Given_N' and 'Gen' to narrow and verify the translated possible Hanzi words.

In this section, we use the mentioned example Pinyin name 'Yuan, Tian-gang' to explain the methods in the second step of LT-NTM. The example name 'Yuan, Tian-gang' has a two-word Given name, 'Tian'

and 'Gang'. In LT-NTM, we process a list of indexed Given names by ordering left to right. Therefore, we separate the explanation of translating the Given name 'Tian' (Figure 3) and 'Gang' (Figure 4) in this section.

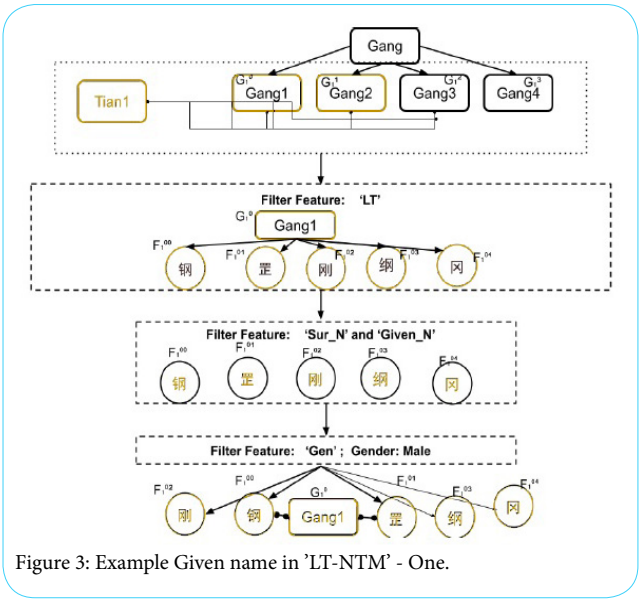

Figure 3: Example Given name in 'LT-NTM' - One.

Figure 3 shows the process of translating the Pinyin Given name 'Tian' of the Pinyin name 'Yuan, Tiangang' to Hanzi characters. We first assign four lexical tones for the proper Pinyin name 'Tian'. And, we pair them with the phoentic name of the translated Surname '袁' and '苑' ('S_i'). The four lexical tones of the proper Pinyin name 'Tian' are 'Tian1'(G_{1^0}), 'Tian2'(G_{1^1}), 'Tian3'(G_{1^2}) and 'Tian4'(G_1^3). Thus, the paired phonetic names are 'Yuan2, Tian1', 'Yuan2, Tian2', 'Yuan2, Tian3' and 'Yuan2, Tian4'.

Next, we apply the filter feature of 'LT' to narrow the quantity of the paired phonetic names (S_i,G_{1^j}). Here, we use regular phonetic patterns that we mentioned in section *regular phonetic patterns and lexical tones* for the process of narrowing down. The method is shown below,

$$P_{(S_i,G_1{}^j)} = \frac{(S_i, G_1{}^j)}{\sum_{\varepsilon=1,j=1}^{\varepsilon\times j}(S_i, G_1{}^j)} \tag{6}$$

$$sd = \sqrt{\frac{\sum\left(P_{(S_i,G_{1^j})} - \sum\left(P_{(S_i,G_{1^j})}\right)\right)^2}{\varepsilon\times j}} \tag{7}$$

$$P_{(S_i,G_1{}^j)} = \begin{cases} Max\left(P_{(S_i,G_1{}^j)}\right), & sd \geq 0.3 \\ 0.35 \leq P_{(S_i,G_1{}^j)}, & 0.0 < sd < 0.3 \end{cases} \tag{8}$$

Here, ε is the quantity of the phonetic Pinyin Surname Si. In formula 6, '(S_i, G_{1^j})' is the frequency of the phonetic Pinyin name '(S_i, G_{1^j})' from the built dataset 'Example lexical tones' (Experimental setup). Through formula 6, we get a list of results '$P(S_i, G_1^j)$'. We then use Standard Deviation (SD) to narrow the quantity of a list of phonetic Pinyin names (S_i, G_1^j) (Formula 7). Here, we generate a filter area for the results of 'sd' (Formula 8). If the result of sd is over 0.3, we pick the maximum result from a list of $P(S_i, G_1^j)$. And, G_1^j in this maximum result is the phonetic Pinyin name of Pinyin Given name g_1^2. If the result of sd is between 0.0 and 0.3, the selected $P(S_i,G_1^j)$ should equal or be larger than 0.35. Here, G_1^j is from the selected $P(S_i,G_1^j)$, and it is the phonetic Pinyin name of Pinyin Given name g_1.

In figure 3, we use the filter feature of 'LT' to narrow the paired phonetic Pinyin names of 'Yuan Tian'. And, 'Yuan2, Tian1' and 'Yuan2, Tian2' are the picked results using regular phonetic patterns. Therefore, 'Tian1' and 'Tian2' are the phonetic Pinyin words of the Pinyin Given name 'Tian'.

We then generate the possible Hanzi words of the phonetic Pinyin words of the Pinyin Given name g_2^1. They are displayed as $F_{l^{j,k}}$(k ∈ {0,1,2, ...,*w*}). In $F_{l^{j,k}}$, *l* is the index number of a list of Pinyin Given names, j is the index lexical tone, k is the index possible Hanzi word of the indexed lexical tone of the Pinyin Given name. In figure 3, the possible Hanzi words of phonetic Pinyin name 'Tian1' are '天' and '添'. And, the possible Hanzi words of phonetic Pinyin name 'Tian2' are '田', '甜', '恬' and '湉'.

Next, we use the filter features of 'Sur_N' and 'Given_N' to narrow and verify these generated possible Hanzi words of the Pinyin Given name 'Tian'. The method is shown below,

$$SP_{F_l^{J,K}} = \frac{S_{F_l^{j,k}}}{\left(S_{F_l^{j,k}} + G_{F_l^{j,k}}\right)} \tag{9}$$

$$GP_{F_l^{J,K}} = \frac{G_{F_l^{j,k}}}{\left(S_{F_l^{j,k}} + G_{F_l^{j,k}}\right)} \tag{10}$$

$$SP_{F_l^{J,K}} < \mathrm{GP}_{\mathrm{F_l^{J,K}}} \Rightarrow F_l^{j,k} \tag{11}$$

Here, formula 9 is the method of the filter feature of 'Sur_N'. Formula 10 is the method of the filter feature of 'Given_N'. In formula 9, '$S_{F_l^{j,k}}$' is the frequency of a possible Hanzi word as a Surname (Experimental setup). In formula 10, '$G_{F_l^{j,k}}$' is the frequency of a possible Hanzi word as a Given name (Section 5.5.1). In LT-NTM, we set a comparison between the results of '$SP_{F_l^{J,K}}$' and 'GPFJ,Kl' to narrow and verify the possible Hanzi words. Here, we set that the result of '$\mathrm{GP}_{\mathrm{F_l^{J,K}}}$' should be larger than the result of '$SP_{F_l^{J,K}}$'.

In figure 3, the generated possible Hanzi words of 'Tian' are narrowed down by the filter features of 'Sur_N' and 'Given_N'. And, the results of the narrowed down Hanzi words for the Pinyin Given name 'Tian' are '天', '添', '甜', '恬' and '湉'.

In LT-NTM, if a Pinyin name 'N' has the label of gender, we use the filter feature of 'Gen' with the labelled gender to narrow and verify the generated possible Hanzi words of the Pinyin given name. And, this method is shown below,

$$MP_{F_1^{J,K}} = \frac{M_{F_1^{j,k}}}{G_{F_1^{j,k}}} \tag{12}$$

$$FP_{F_1^{J,K}} = \frac{F_{F_1^{j,k}}}{G_{F_1^{j,k}}} \tag{13}$$

$$\begin{cases} Gender = None, \Rightarrow F_1^{j,k} \\ Gender = Male, MP_{F_1^{J,K}} > FP_{F_1^{J,K}} \Rightarrow F_1^{j,k} \\ Gender = Female, FP_{F_1^{J,K}} > MP_{F_1^{J,K}} \Rightarrow F_1^{j,k} \end{cases} \tag{14}$$

Where ' $G_{F_1^{J,k}}$ ' is the frequency of a possible Hanzi word as a Given name. Here, in formula 12, $M_{F_1^{J,k}}$ is the frequency of a possible Hanzi word as a Given name in male (Section 5.1). And, ' $MP_{F_1^{J,K}}$' is the result of the possible Hanzi word as a Given name in male. In formula 13, '$F_{F_1^{J,k}}$' is the frequency of a possible Hanzi word '$F_{F_1^{J,k}}$' as a Given name in female (Section 5.1). And, '$FP_{F_1^{J,K}}$' is the result of the possible Hanzi proper name ' $F_{F_1^{J,k}}$ ' as a Given name in female.

In formula 14, we set three conditions to judge the results from the filter feature of 'Gen' with possible Hanzi words $F_{F_1^{J,k}}$. If the labelled gender of Pinyin name '*N*' is *None,* we do not continue to process the possible Hanzi words $F_{F_1^{J,k}}$ with the filter feature of '*Gen*'. If the labelled gender of Pinyin name '*N*' is Male, the result of $MP_{F_1^{J,K}}$ should be larger than the result of $FP_{F_1^{J,K}}$. If the labelled gender of Pinyin name '*N*' is Female, the result of $FP_{F_1^{J,K}}$ should be larger than the result of $MP_{F_1^{J,K}}$.

In figure 3, the labelled gender of Pinyin name 'Yuan, Tian-gang' is male. We use the filter feature of 'Gen' to narrow the generated possible Hanzi words of Pinyin Given name 'Tian'. The translated Hanzi characters of Pinyin Given name 'Tian' are '天' and '添'.

As the example name 'Yuan, Tian-gang' has a two-word Given name, we will explain the process of translating the second word Pinyin Given name 'Gang' in LT-NTM. Figure 4 shows the process of translating the Pinyin Given name 'Gang' to Hanzi character.

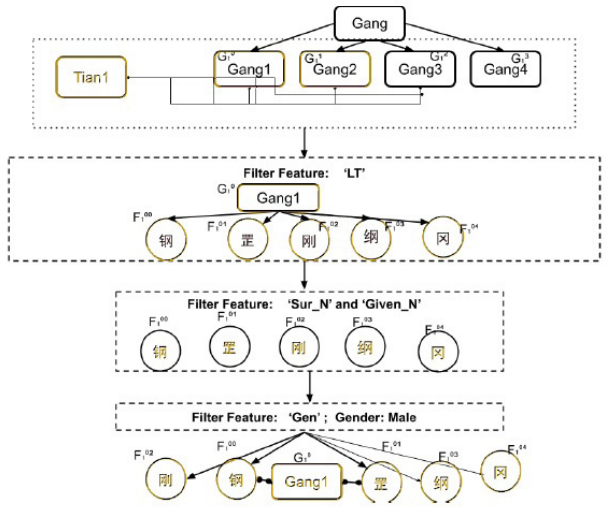

Figure 4: Example Given name in 'LT-NTM' - Two.

In figure 4, we first pair the phonetic name of the translated Hanzi characters of the Pinyin Given name 'Tian' with the generated possible phonetic words of the Pinyin Given name 'Gang'. We then use the filter feature 'LT' to select the paired phonetic names of 'Tian Gang'. Here, 'LT-NTM' selects 'Tian1 Gang1' as the suitable pair of Pinyin Given name. Therefore, 'LT-NTM' uses the phonetic proper name 'Gang1' to generate the possible Hanzi words of 'Gang'. And, '钢', '罡', '刚', '纲' and '冈' are the generated possible Hanzi words of 'Gang'. Next, we use the filter features of 'Sur_N' and 'Given_N' to verify these generated possible Hanzi words of 'Gang'. Here, '钢', '罡', '刚', '纲' and '冈' are the results by using the filter features of 'Sur_N' and 'Given_N'. In the end, we use the filter feature of 'Gen' to verify these possible Hanzi words. Here, the labelled gender of 'Yuan,Tian-gang' is male. Therefore, the results of translating Pinyin 'Gang' to Hanzi characters are '钢', '罡', '刚', '纲' and '冈'.

Data for evaluation

In the evaluation of this study, we build three data sets for training and testing 'LT-NTM'. To assist building these three data sets, we use three open data sources. The first data set is called 'Chinese Name Database 1930-2008' which is created by Bao HWS. [19] from the National Citizen Identity Information Center (NCIIC) of China in 2008. It covers 1,163,760 Han Chinese population [19]. In 'Chinese Name Database 1930-2008', we apply the corpus of 'familyname' (Also referred to as 'Surname') and the corpus of 'givenname' (Also referred to as 'Given name') as our training data. The second data set is called 'Chinese Names Corpus' which is created by NameMoe [20]. This corpus includes 1,163,760 names in Hanzi, which are extracted from a billion names database. In our evaluation, we apply this database for model training and testing. The last corpus is called '9800 Chinese Names with Gender' which is created by Bao HWS [21]. This data set includes 4,900 Chinese male names and 4,900 Chinese female names and their labelled genders. In our evaluation, we apply this data set for model training and testing.

To help build our data sets, we randomly distribute these data sets to our evaluations training and testing data. We randomly picked 581,880 data from the data set of 'Chinese Names Corpus' to be the training data of 'LTNTM'. The left 581,880 data in 'Chinese Names Corpus' is arranged to be the testing data for 'LT-NTM'. We then picked 4,900 data from the data set of '9800 Chinese Names with Gender' to be the training data of 'LT-NTM', and the left 4,900 data to be the testing data of 'LT-NTM'.

Training data setup

As we mentioned at the beginning of this section, we use three online data sets for 'LT-NTM' model training. These three data sets are 'Chinese Name Database 1930-2008', 'Chinese Names Corpus' and '9800 Chinese Names with Gender'.

To manage these open data sources, we use the data set of 'Chinese Name Database 1930-2008' to build the training data for 'LT-NTM'. Moreover, we apply the data sets of 'Chinese Names Corpus' and '9800 Chinese Names with Gender' to build the training data set on regular phonetic pattern training for 'LT-NTM'. In 'Chinese Name Database 1930-2008', we collected the data sets of 'familyname' (Also referred to as' Surname') and 'givenname' (Also referred to as' Given name') to build for our first training data set. Here, we collected 4,420 Hanzi characters from 'Chinese Name Database 1930-2008'. We first got 1,806 Hanzi characters whose element is 'n.1930_2008' (Frequency of the character between 1930 to 2008) from the 'familyname' data set. We then got 2,614 Hanzi characters whose elements are 'Pinyin', 'n.male' and 'n.female' from the data set of 'givenname'. In the end, we combined these collected Hanzi characters to build our first training data set. We call it 'Training data-Set'. Table 3 displays an example data from our first training data set 'Training data-Set'.We set five elements for our models to understand the data from the training data set of 'Training data-Set'.

Our second training data set is used for regular phonetic pattern training in 'LT-NTM'. We call it 'Example lexical tones'. We collected the data from two open data sources, 'Chinese Names Corpus' and '9800 Chinese Names with Gender', to build 'Example lexical tones'. We collected 586,780 Hanzi names from these two open data sources to build the 'Example lexical tones' training data set. On building the 'Example lexical tones' training data set, we set two steps to process

the collect names for building 'Example lexical tones'. We first split a Hanzi name by two Hanzi characters. For example, in table 4, a three-word name '班冬冬' can be split as '班 冬' and '冬冬'.

We then transfer the split Hanzi characters into phonetic pinyin characters and statistical their frequency. Table 4 presents the phonetic Pinyin data from three Example Hanzi names for the 'Example lexical tones' training data set. For Example, the three-word Hanzi name '班冬冬' is processed as 'ban1,dong1' and 'dong1,dong1' in the 'Example lexical tones' training data set. Here, 'ban1,dong1' is the phonetic character of the split name of '班冬'. And, 'dong1,dong1' is the phonetic character of '冬冬' in '班冬冬'.

Testing data setup

In our evaluation, we collected the data from the open data sources of 'Chinese Names Corp' and '9800 Chinese Names with Gender' to be our testing data.

Table 5 displays the testing data assignment of the collected 44,172 data from the open data sources.

We built three testing data sets by the lengths of Hanzi names. They are 'Two_word' data set, 'Three_word' data set and 'Four_word' data set (Table 5).

Table 6 displays the testing data examples from our testing data sets. For example, in our testing data set, a three-word Pinyin name

Hanzi	Pinyin	Surname	Given Name (Male)	Given Name (Female)
麒	qi	0	232	19

Table 3: An Example From 'Training data-Set'.

Hanzi Name	Proceeding	Phonetic Data in 'LT-NTM'
艾吉	艾吉	ai4, ji2
班冬冬	班冬; 冬冬	ban1,dong1 ; dong1,dong1
欧阳智山	欧阳; 阳智;智山	ou1,yang2;yang2,zhi4;zhi4,shan1

Table 3: An Example From 'Training data-Set'.

Data set	Label	Amount (Data)
Two_word	Male	9,920
	Female	9,163
Three_word	Male	12,684
	Female	12,377
Four_word	Male	12
	Female	16

Table 5: Information of Testing Data Assignment.

'Ou Yang Shan' is processed as 'Ou-Yang, Shan'. Here, 'Ou-Yang' is the labelled Pinyin Surname of 'Ou-Yang, Shan'. And, 'Shan' is a one-word Pinyin Given name. The symbol of 'Ou-Yang, Shan' in our experiment is 'g_1, g_1^2'. Therefore, we set this name as a two-word name and assigned it into the 'Two-word' testing data set.

Model training

Parameters in 'LT-NTM'

To help processing the filter feature of 'LT', we focus on training the parameters of 'SD' and 'V' (Formula 7 and 8). Here, 'V' is a filter area for the results of a paired phonetic names, such as the filter area of ' $P(S_i, G_{1j})$' and 'sd' (Formula 8). Additionally, 'SD' is the amount of 'Standard deviation' in the method of regular phonetic patterns of 'LT-NTM' (Formula 7). Therefore, we use 3,000 testing data from our testing data set (Experimental setup) on the model training of parameters. Table 7 reports the results of this model training.

In the table, 'Acc' is the accuracy of the hypothesis of different parameters of 'SD' and 'V'. And, 'Ave(Pos-N)' is the average number of the translate Hanzi characters of a Pinyin proper name in a Pinyin name 'N'. Depending on the model training report results, we set '*0.6' as the value for the parameter of 'SD'. And, '0.3' as the value for the parameter of 'V'.

Four-word name hypothesis set

To help to translate a four-word Pinyin name to Hanzi characters, we build a hypothesis set to train LT-NTM.

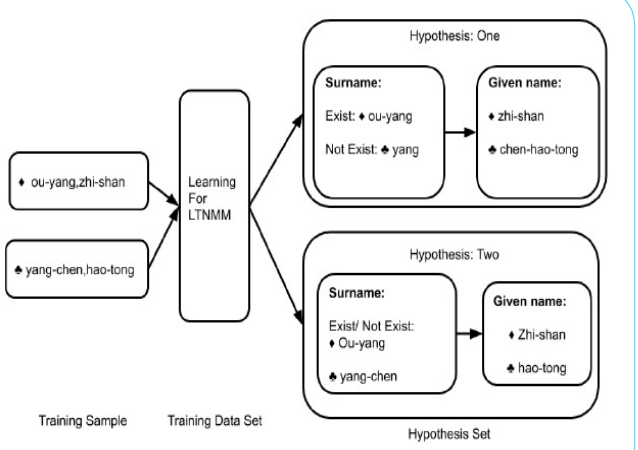

Figure 5: The Hypothesis Set of 'LT-NTM' For Four Word Names

Normally, we separate a four-word Mandarin name into a two-word Surname and a two-word Given name for name identification. However, some collected names' 'Surname' can not be found in the professional dictionaries. Therefore, we propose two candidate LT-NTM methods to approach a higher name identification accuracy. In this model training experiment, we focus on four-word name identification. Figure 5 displays the process of two candidate LT-NTM methods for separating Surnames. In the figure, we use two four-word example names to indicate two hypotheses. Two example Pinyin names are 'ou-yang, zhi-shan' and 'yang-chen, hao-tong'. And, two candidate LT-NTM methods are named 'Hypothesis: one' and 'Hypothesis: Two'. In figure 5, we use ♦ to present the example Pinyin name 'ou-yang, zhi-shan'. And, we use ♣ to present the example Pinyin name 'yang-chen, hao-tong'. In 'Hypothesis: one', the Surname and the Given name of a Pinyin are separated by using our 'Learning For LTNTM' training data set (Experimental setup). Here, we classify the indexed first word as the surname of a four-word name when the two-word Surname of a Pinyin name cannot be identified. In 'Hypothesis: Two', the Surname and the Given name of a Pinyin name are separated

using the 'Learning For LTNTM' training data set. Here, we classify the indexed first word and second word as the surname of a four-word name when the two-word Surname of a Pinyin name cannot be identified.

Pinyin name	Testing Data	Transfer to Symbol
Ai Ji	Ai, ji	g_1, g_1
Ou Yang Shan	Ou-Yang, Shan	g_1, g_1
Ban Dong Dong	Ban, dong-dong	$g_1, g_2^1 - g_2^2$
Ou Yang Zhi Shang	Ou-Yang, Zhi-Shang	$g_1, g_2^1 - g_2^2$
Zhong Li Juan Ren	Zhong, li-juan-ren	$g_1, g_2^1 - g_2^2 - g_2^3$

Table 6: Testing Data Examples.

SD	V	Acc(%)	Ave(Pos-N)
*0.3	> 0.2	61.33	5.08
	> 0.3	57.96	5.14
	> 0.4	53.10	4.67
*0.4	> 0.2	62.96	5.73
	> 0.3	58.10	5.15
	> 0.4	52.20	4.67
*0.5	> 0.2	63.20	5.75
	> 0.3	58.10	5.71
	> 0.4	53.10	4.67
*0.6	> 0.2	63.20	5.75
	> 0.3	**60.76**	**5.15**
	> 0.4	53.10	4.67
*0.7	> 0.2	63.20	5.75
	> 0.3	58.10	5.15
	> 0.4	53.10	4.67

Table 7: Model training of 'SD' Range and 'V' Range in LT-NTM/

data set (*Experimental setup*). Here, we classify the indexed first word as the surname of a four-word name when the two-word Surname of a Pinyin name cannot be identified. In 'Hypothesis: Two', the Surname and the Given name of a Pinyin name are separated using the 'Learning For LTNTM' training data set. Here, we classify the indexed first word and second word as the surname of a four-word name when the two-word Surname of a Pinyin name cannot be identified.

In figure 5, two example Pinyin names are identified by using the 'Learning For LT-NTM' training data set. Here, LT-NTM separates the example Pinyin names into Surnames and Given names for further process. In the example name 'ou-yang, zhi-shan' (♦), 'ou-yang' is a generated Surname that is identified by the professional dictionary. Therefore, in LT-NTM, the results of the identification of Surname and Given name of Pinyin name 'ou-yang, zhi-shan' are the same on both 'Hypothesis: one' and 'Hypothesis: Two'. Here, 'ou-yang' is identified as a surname of 'ou-yang, zhi-shan'. And, 'zhi-shan' is identified as a Given name of Pinyin name 'ou-yang, zhi-shan'. For processing the example name of 'yang-chen, hao-tong' (♣), 'yang-chen' cannot be iden identified as a surname by using the professional dictionary. Figure 5 shows two different distributions of 'yang-chen, hao-tong' between 'Hypothesis: one' and 'Hypothesis: Two'. In 'Hypothesis: one', 'yang' is identified as a Pinyin Surname and 'chen-hao-tong' is identified as the surname of 'yang-chen, hao-tong', and the Given name of 'yang-chen, hao-tong' is identified as 'hao-tong'.

Table 8 shows the distribution of two example names on identifying the Hanzi words for the first word of the Pinyin Given names. In the table, Pinyin name 'ou-yang, zhi-shan' (♦) has the same distribution in both 'Hypothesis: one' and 'Hypothesis: Two'. Therefore, to translate the Given name of 'zhi' in LT-NTM, we pair the phonetic name of 'zhi' with the translated Hanzi versions of the Pinyin Surname 'ou-yang'. In table 8, in 'Hypothesis: One', the identified surname of the four-word Pinyin name 'yang-chen, hao-tong' is 'yang'. And, the identified first word of the Given name of Pinyin name 'yangchen, hao-tong' is 'chen'. In table 8, in 'Hypothesis: Two', 'yang-chen' is the identified Surname of 'yangchen, hao-tong'. And, 'hao' is identified first word of the Given name of 'yang-chen, hao-tong'. Therefore, in LTNTM, we pair the phonetic names of translated Surname 'chen'(S_i) with the phonetic Pinyin words of 'hao' for translation.

In the experiments, we use 28 four-word names from the training sample to evaluate this Hypothesis set. We evaluate the Hypothesis set by using labelled gender and unlabeled gender. Table 9 shows the accuracy of each Hypothesis. In table 9, the results show that the accuracy of the Hypothesis set without using 'gender' is higher than the Hypothesis set using 'gender'. We think the reason is that the unlabelled translated names have more comprehensive than the labelled translated names. The accuracy of 'Hypothesis: One' is higher than the accuracy of 'Hypothesis: Two' when both of them are considered without using the label of 'gender'. Therefore, we use 'Hypothesis: One' as the final Hypothesis model for 'LT-NTM'.

Hypothesis Set	Example Name 'N'	S_i	G_i^j
Hypothesis: One	♦ ou-yang, zhi-shan	ou-yang	zhi
	♣ yang-chen, hao-tong	yang	chen
Hypothesis: Two	♦ ou-yang, zhi-shan	ou-yang	zhi
	♣ yang-chen, hao-tong	chen	hao

Table 8: Example Names in the Hypothesis set.

Hypothesis Set	Data Amount	Acc(%)
Hypothesis: One	No Gender (28)	10.71
	Male (12)	8.33
	Female (16)	0.00
Hypothesis: Two	No Gender(28)	**17.85**
	Male(12)	8.33
	Female(16)	12.50

Table 9: Results of the Model Training of Four-word Name Hypothesis Set.

Experimental setup

In the experiments of 'LT-NTM', we build a method to analyse the result data of different lengths of Hanzi names. Our testing data includes three different lengths of Mandarin names. They are two-word name, threeword name and four-word name. The method of analysing the translation results of these three different lengths of names is shown as follows,

$$\begin{cases} N = (g_1, g_2^1) \; and \; len(l) = 1 \Rightarrow [C_i!, F_1^j] \\ N = (g_1, g_2^1) \; and \; len(l) = 1 \Rightarrow [C_i!, F_1^j;, F_2^j] \\ N = (g_1, g_2^1) \; and \; len(l) = 1 \Rightarrow [C_i!, F_1^j;, F_2^j :, F_3^j] \end{cases} \quad (15)$$

Here, at the left of '⇒' are the Pinyin names. And, at the right of '⇒' are the translated Hanzi names of the Pinyin names. In the evaluation, we design three different translated Hanzi name styles for three different lengths of Mandarin names. Therefore, in formula 15, '*len(l)*=1' means that a Mandarin name has a one-word Given name. Here, we set $[C_i!, F_1^j]$ as the output of the translated Hanzi characters of a two-word Pinyin name g_1, g_2^1. And, in $[C_i!, F_1^j]$, we use '!' as a sign to identify the translated Hanzi surname of 'g_1, g_2^1'.

The path of the inheritance graph

In formula 15, '*len(l)* = 2' means that a Pinyin name has a two-word Given name. Here, we set $[C_i!, F_1^j; ,F_2^j]$ as the output of the translated Hanzi words of a three-word Pinyin name 'g$_1$, g$_2^1$, g$_2^2$'. And, in $[C_i!, F_1^j; , F_2^j]$, we use '!' as a sign to identify the translated Hanzi surname of a name. ;' is a sign used to identify the translated Hanzi words of the first word of a Pinyin given name.

In formula 15, '*len*(*l*) = 3' means that a Pinyin name has a three-word Given name. Here, we use $[C_i!, F_1^j; , F_2^j : , F_3^j]$ as the output of the translated Hanzi words of a four-word Pinyin name g_1, g_2^1, g_2^2, g_2^3. In $[C_i!, F_1^j; ,F_2^j :, F_3^j]$, we use ':' as a sign to identify the translated Hanzi words of the second word of a Given name. Besides, in a translated four-word name, we use '?' to indicate the translated Hanzi words of the second word of a twoword surname.

Table 10 displays the examples of three different lengths of Mandarin names and their translation results by using LT-NTM. In the table, '*len*(*l*)' means the length of a Given name in a name.

Experiment Results

To test the proposed 'LT-NTM', we compare it with 'Google Translate'. We use 44,172 data from the open data source (Experimental setup) to test our model. In the evaluation, we use the Chi-Square test as one of our testing methods, with the Statistical Significance level be equal to 0.05. We also use Accuracy (Acc) to report model's performance using different lengths of people's names.

Table 11 indicates the evaluation results of 'LT-NTM' and 'Google Translate'. In this experiment, we set the null hypothesis for the Chi-Squared test as 'LT-NTM' and 'Google Translate' have similar accuracy in translating Pinyin names to Hanzi characters. In table 11, two P-values are smaller than 0.05 when' LT-NTM' and 'Google Translate' translate the Mandarin names in Two words and Three words. Therefore, there is a statistically significant relationship between' LT-NTM' and 'Google Translate' on translating two-word and threeword names. However, P-value is larger than 0.05 when 'LT-NTM' and 'GoogleTranslate' translate four-word names. We think the reason is that there is limited test-ing data in four-word names. Based on this, the accuracy of 'LT-NTM' is 69.02%, the accuracy of 'Google Translate' is 30.00%.

Pinyin Name	Hanzi Name	*len(l)*	Result
Zou, si	邹,思	1	['邹!, '斯', '丝', '', '思', '锶', '鸶']
Ai, hu-sheng	艾, 虎生	2	['艾!', '琥;', '浒;', '虎;', '笙', '升', '生', '声']
Zhong, li-juan-ren	钟, 李隽仁	3	['钟!', '种!', '砺;', '雳;', '立;'', '隶;', '励;', '笠;', '力;', '人','仁']

Table 10: Examples of different lengths of names in the 'LT-NTM' Experiment.

ing data in four-word names. Based on this, the accuracy of 'LT-NTM' is 69.02%, the accuracy of 'Google Translate' is 30.00%.

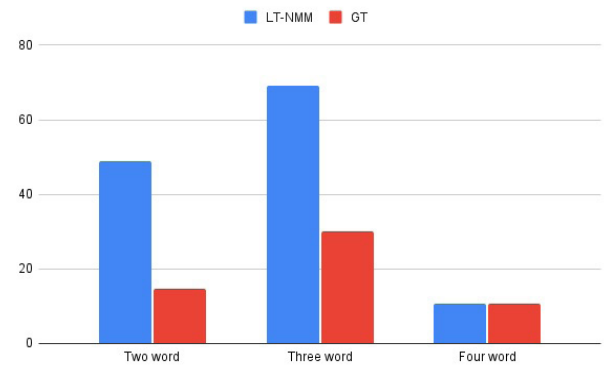

Figure 6: The Accuracy of 'LT-NTM' and 'Google Translate'(GT)

Figure 6 shows the accuracy of 'LT-NTM' and 'Google Translate' which referred table 11. In figure 6 , 'LTNTM' has better performance of Mandarin name translation than 'Google Translate'. On translating different lengths of Mandarin names, 'LT-NTM' and 'Google Translate' are all doing well on three-word names.

However, the ranking of these three methods when looking at the average amounts of the translated Hanzi characters for each proper Pinyin name in Mandarin is opposite (Table 12). In table 12, '(Length)' means the average amount of the translated Hanzi characters of each proper Pinyin name, where shorter length means fewer resulting characters which is preferable. 'Google Translate' has the shorter average amount on each translated result than 'LT-NTM'. 'LT-NTM' has better performance in translating two-word names. In this experiment, 'Google Translate' is consistent and performs well when considering the average amount of the translated result of each proper Pinyin name.

Data Set	LT-NTM (Acc(%))	GT (Acc(%))	*P* value
Two_word Name	48.79	14.74	< 0.0001
Three_word Name	**69.02**	**30.00**	< 0.0001
Four_word Name	10.71	10.71	0.50

Table 11: Comprehension Name Correct Response Rate of 'LT-NTM' and Google Translate (GT).

Data Set	LT-NTM (Length)	GT (Length)
Two_word Name	**5.28**	**2**
Three_word Name	5.74	2
Four_word Name	7.09	2

Table 12: The Average Amount of the translated result of each proper Pinyin name in DT-NTM, LT-NTM, and Google Translate(GT).

Conclusion

This article has implemented a novel model for translating Mandarin names, 'LT-NTM'. Our methods relied on the labelled Pinyin names and their Hanzi characters as our training and testing data. We then trained our designed model. Finally, we applied the data sets to

testing our novel model and comparing it with Google Translate. Taken together, our study has provided a novel method on people's name translation. And we have pointed out that this model has a good performance on people's name translation in Mandarin name, which has better accuracy than Google translate. In the future, we want to extend our model so that it can have better performance when translating the proper names in more languages.

Competing Interests

The author declare that he has no competing interests.

References

1. Study.com, "Teaching students to select diverse texts."
2. Pour BS (2009) How to translate personal names. Translation Journal 3: 1-13.
3. Smarr J, Manning CD (2002) Classifying unknown proper noun phrases without context," Technical Report 2002-46, Stanford InfoLab, April 2002.
4. Cohen AA, Ravikumar P, Fienberg SE (2003) A comparison of string distance metrics for namematching tasks," in Proceedings of the 2003 International Conference on Information Integration on the Web, IIWEB'03, p. 73-78, AAAI Press.
5. Peng N, Yu M, Dredze M (2015) An empirical study of Chinese name matching and applications," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), (Beijing, China), pp. 377–383, Association for Computational Linguistics.
6. n. anatomy, "Oed online,oxford university press," December 2020. Accessed: 2021-01.
7. Zhao H, Kamareddine F (2019) A novel phonetic algorithm for predicting chinese names using chinese pin yin. in MLDM 1: 78-92.
8. Pym A (2004) The moving text: localization, translation, and distribution, vol. 49. John Benjamins Publishing.
9. Ordudari M (2007) Translation procedures, strategies and methods. Translation journal 11: 8.
10. Särkkä H (2007) Translation of proper names in nonfiction texts. Translation journal 11: 1.
11. Sand V (2021) Translation or rewriting of proper names: A study of children's literature across a century. Linnaeus university, Sweden.
12. Wan S, Verspoor CM (198) Automatic English- Chinese name transliteration for development of multilingual resources," in 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2, (Montreal, Quebec, Canada), pp. 1352–1356, Association for Computational Linguistics.
13. Newmark P (1988) A textbook of translation, Prentice hall New York, vol. 66.
14. Lin D, McBride-Chang C, Shu H, Zhang Y, Li H, etal. (2010) Small wins big: analytic pinyin skills promote Chinese word reading. Psychol Sci 1: 1117-1122.
15. Chung KKH (2002) Effective use of hanyu pinyin and english translations as extra stimulus prompts on learning of chinese characters. Educational Psychology 22: 149-164.
16. Institute of Linguistics, Chinese Academy of Social Sciences, Xinhua Dictionary. Beijing: The Commercial Press, 10th edition ed., 2004.
17. Wang WC, Chen HC, Ji ZK, Hsiao HI, Chiu YS, Ku LW (2016) Whose nickname is this? recognizing politicians from their aliases," in Proceedings of the 2nd Workshop on Noisy Usergenerated Text (WNUT), (Osaka, Japan), pp. 61–69, The COLING 2016 Organizing Committee.
18. Liu L, Peng D, Ding G, Jin Z, Zhang L, et al. (2006) Dissociation in the neural basis underlying Chinese tone and vowel production. Neuroimage 15: 515-523.
19. Bao HWS, Chinesenames: Chinese name database 1930-2008.
20. Fei Y (2019) Chinese names corpus.
21. Bao HWS, 9800 chinese names with gender.