An Approach of Using the Web as a Live Corpus for Spoken Transliteration Name Access

Ming-Shun Lin, Chia-Ping Chen, Hsin-Hsi Chen***

Department of Computer Science and Information Engineering
National Taiwan University, Taipei 106
TAIWAN
{d91022 , **hhchen}@csie.ntu.edu.tw

Department of Computer Science and Engineering
National Sun Yat-Sen University
70, Lien-Hai Road, Kaohsiung, Taiwan 804
*cpchen@cse.nsysu.edu.tw

Abstract

Recognizing transliteration names is challenging due to their flexible formulation and coverage of a lexicon. This paper employs the Web as a huge-scale corpus. The patterns extracted from the Web are considered as a live dictionary to correct speech recognition errors. In our approach, the plausible character strings recognized by ASR (Automated Speech Recognition) are regarded as query terms and submitted to Google. The top *n* returned web page summaries are entered into PAT trees. The terms of the highest scores are selected. Total 100 Chinese transliteration names, including 50 person names and 50 location names, are used as test data. In the ideal case, we input the correct syllable sequences, convert them to text strings and test the recovery capability of using Web corpus. The results show that both the recall rate and the MRR (Mean Reciprocal Rank) are 0.94. That is, the correct answers appear in the top 1 position in 94 cases. When a complete transliteration name recognition system is evaluated, the experiments show that ASR model with a recovery mechanism can achieve 3.82% performance increases compared to ASR only model on character level. Besides, the recovery capability improves the average ranks of correct transliteration names from the 18th to the 3rd positions on word level.

1. Introduction

Named entities [1], which denote persons, locations, organizations, etc., are common foci of searchers. Capturing named entities is challenging due to their flexible formulation and up-to-date use. The issues behind speech recognition make named entity recognition more challenging on spoken level than on written level. This paper emphasizes on a special kind of named entities, called transliteration names. They denote foreign people, places, etc. Spoken transliteration name recognition is useful for many applications. For example, cross language image retrieval via spoken query aims to employ spoken queries in one language to retrieve images with captions in another language [2].

In the past, Appelt and Martin [3] adapted TextPro system for processing text to processing transcripts generated by a speech recognizer. Miller et al [4] analyzed the effects of out-of-vocabulary errors and loss of punctuation in name finding of automatic speech recognition. Huang and Waibel [5] proposed an adaptive

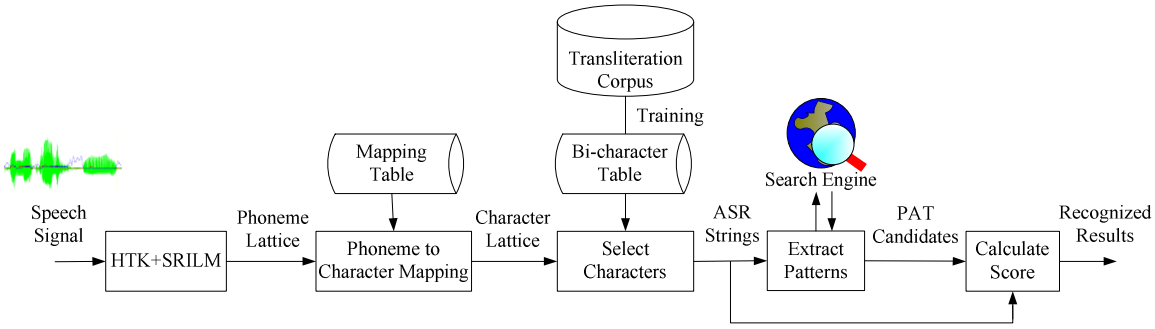

Figure 1. Flow of transliteration name recognition.

method of named entity extraction for meeting understanding. Chen [6] dealt with spoken cross-language access to image collection. The coverage of a lexicon is one of the major issues in spoken transliteration name access. Recently, researchers are interested in exploring the Web, which provides huge-collection of up-to-date data, as a corpus. Keller and Lapata [7] employed the Web to obtain frequencies for bigrams that are unseen in a given corpus.

In this paper, we consider the Web as a live dictionary for recognizing spoken transliteration names, and employ fuzzy search capability of Google to retrieve relevant web page summaries. Section 2 sketches the overall flow of our method. Section 3 employs PAT trees to learn patterns from the Web dynamically and correct the recognition errors. Section 4 shows the experiments with/without the uses of the Web. Section 5 concludes the remarks.

2. Flow of transliteration name recognition

A spoken transliteration name recognition system shown in Figure 1 accepts a speech signal denoting a foreign named entity, and converts it into a character string. It is composed of the following four major stages. Stages (1) and (2) are fundamental tasks of speech recognition. Stages (3) and (4) try to correct the speech-to-text errors by using the Web.

(1) At first, we employ the speech recognition models built by HTK (http://htk.eng.cam.ac.uk/) and SRILM (http://www.speech.sri.com/projects/srilm/) toolkits to get a syllable lattice of a speech signal.

(2) Then, the syllable lattice is mapped into a character lattice using a mapping table. Top-n character strings are selected from the character lattice using bi-character model trained from a transliteration name corpus. The character strings are called ASR strings hereafter.

(3) Next, each ASR string is regarded as a query, and is submitted to a web search engine like Google. From the top-m search result summaries of a query (i.e., an ASR string), the higher frequent patterns similar to the ASR string are considered as candidates. Because we employ PAT tree [8, 11] to extract patterns, the patterns are called PAT candidates hereafter. For PAT tree example, “湯姆漢克斯湯姆克魯斯喬治克魯尼” with MS950 encode, be shown in Figure2. The circle represents semi-infinite string number. The number located over the circle is length. The length indicates the first different bit of the character strings recorded in the sub-trees. In this example, the highest length patterns are “克魯” and “湯姆” on the nodes (7, 12) and (0, 5)

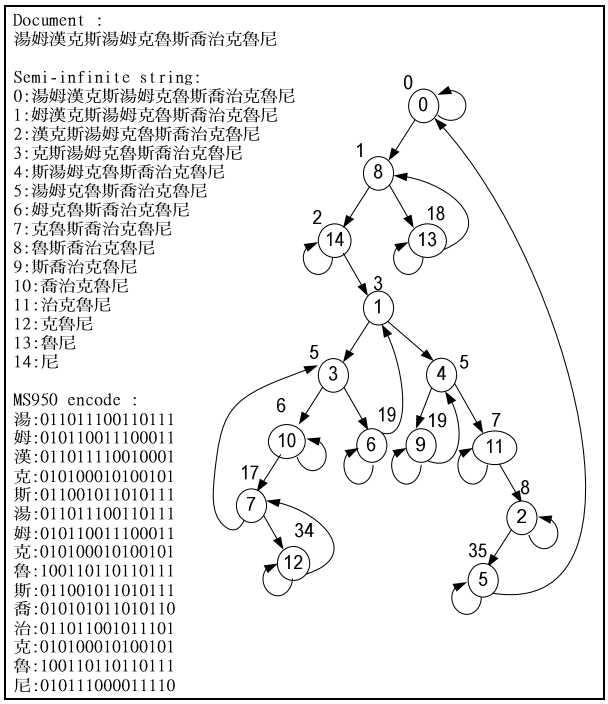

Figure 2. An example for extracting highest length patterns and its frequency.

with length 33 and 34 bits. The second highest length patterns are “克”,”魯”,”姆” and “斯” on nodes (3, 7, 12),(8, 13),(1, 6) and (4, 9) with length 16,17,18 and 18 bits. The pattern extraction task will be discussed in detail in Section 3.

(4) Finally, the PAT candidates of all ASR strings will be merged together, and ranked by their number of occurrences and similarity scores. Candidates of the best ranks will be regarded as recognition results of a spoken transliteration name.

Consider an example shown in Figure 3. The Chinese speech signal is a transliteration name “湯姆克魯斯” in Chinese denoting a famous movie star “Tom Cruise”. Syllable lattice illustrates different combinations of syllables. Each syllable corresponds to several Chinese characters. For example, ke is converted to “克”, “柯”, “科”, “可”, “喀”, “刻”, etc. ASR strings “塔莫克魯斯”, “塔門克魯斯”, “塔莫柯魯斯”, etc. are selected from character lattice. Through Google fuzzy search using query “塔莫克魯斯”, some summaries of Chinese web pages are reported in Figure 4. Although common transliteration of “Tom Cruise” in Chinese is “湯姆克魯斯”, which is different from the query “塔莫克魯斯”, fuzzy matching by Google can still identify the relevant summaries containing the correct transliteration. We call this operation recognition error recovery using the Web hereafter.

In the above examples, partial matching part is enclosed in rectangle symbol, e.g., “克魯斯”, and the correct transliteration name is underlined, e.g., “湯姆克魯斯”. Summaries (1), (4) and (5), mention a movie star “湯姆克魯斯” (Tom Cruise), and summaries (2) and (3) mention a football star “克魯斯” (Cruz). Figure 3 shows that the PAT patterns like “聖塔克魯斯”, “湯姆克魯斯”, “姆克魯斯演”, etc. are proposed. After merging and ranking, the possible recognition results in sequence are “湯姆克魯斯”, “洛普克魯茲”, “聖塔克魯茲”, etc.

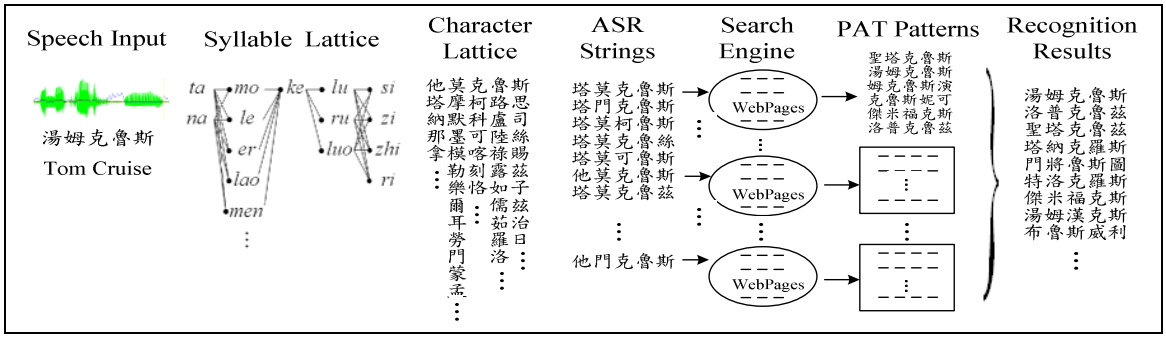

Figure 3. An example for recognizing transliteration name "湯姆克魯斯" ("Tom Cruise").

(1) … 至於贏家部份，則還是湯姆漢克斯、湯姆克魯斯、喬治克魯尼這些老面孔，…

(2) … 第 76 分鐘，克魯斯換下梅開二度的維埃裏。

(3) … 國際米蘭(4-4-2)：豐塔納/科爾多瓦，布爾迪索，馬特拉齊，法瓦利/斯坦科維奇，貝隆，扎內蒂，埃姆雷/克魯斯，馬丁斯。

(4) … 提起妮可即發火湯姆克魯斯"想殺死記者".

(5) … 電影節最具看點的明星當然非妮可基德曼與湯姆克魯斯有望在水城的戲劇性重逢莫屬。

Figure 4. Summaries of fuzzy search for query "塔莫克魯斯"

3. Recognition error recovery using the Web

The error recovery module tries to select the higher frequent pattern from the Web search results, and substitute the speech recognition results of Stages 1 and 2 (shown in Section 2) with the pattern. PAT tree [8,11], which was derived from Patricia tree, can be employed to extract word boundary and key phrases automatically. In this paper, the Web search results of an ASR string will be placed in a PAT tree and PAT candidates will be selected from the tree. Two issues are considered. A PAT candidate should occur more times in the PAT tree and should be similar to the ASR string.

The frequency Freq of a PAT candidate can be computed easily from PAT tree structure. The similarity of a PAT candidate and an ASR string is modeled by edit distance, which is minimum number of insertions, deletions and substitutions to transform one character string (ASR) into another string (PAT). The less the number is, the more similar they are. The similarity Sim of ASR and PAT strings is the length of string alignment minus the number of edit operations.

Finally, the ranking score of a PAT string relative to an ASR string is defined as follows.

$$Score(ASR, PAT) = \alpha \times Freq(PAT) + \beta \times Sim(ASR, PAT)$$

It is computed by weighted merging of the frequency of the PAT string, and the similarity of the ASR string and PAT string. This value determines if the ASR string will be replaced by the PAT string. In the above example, Freq(湯姆克魯斯)=43 and Sim(塔莫克魯斯, 湯姆克魯斯)=3.

4. Experimental results

The speech input to the transliteration name recognition system is Chinese utterance. We employed 51,114 transliteration names [9] to train the bi-character model specified in Section 2. In the experiments, the test data include 50 American state names, 29 movie star names from 31st annual people's choice awards (http://www.pcavote.com), and 21 NBA star names from NBA 2005 all star (http://www.nba.com/allstar2005/). The test set is different from the training set and it is open test. Because there may be more than one transliteration for a foreign named entity, the answer keys are manually prepared and checked with respect to the Web. For example, "Arizona" has four possible transliterations in Chinese – say, "亞利桑納", "亞歷桑納", "亞利桑那", and "亞歷桑那". On the average, there are 1.9 Chinese transliterations for a foreign name in our test set. In appendix A lists the name test set and its answer keys. As shown in Section 2, the transliteration name recognition system is composed of four major stages. Stages 1 and 2 perform the fundamental speech recognition task, and Stages 3 and 4 perform the error recovery task. To examine the effects of these two parts, we evaluate them separately and wholly in the following two subsections.

4.1 Performance of error recovery task

Assume correct syllables have been identified in speech recognition task. We simulate the cases by transforming all the characters in the answer keys back to syllables. Then, Stage 2 maps the syllable lattice into character lattice. Total 50 ASR strings are extracted from character lattice at stage 2, and submitted to Google. Finally, the best 10 PAT candidates are selected. We use recall rate and MRR (Mean Reciprocal Rank) [10] to evaluate the performance. Recall rate means how many transliteration names are correctly recognized. MRR defined below means the average ranks of the correctly identified transliteration names in the proposed 10 PAT candidates.

$$MRR = \frac{1}{M}\sum_{i=1}^{M} r_i \qquad (1)$$

, where $r_i = 1/rank_i$ if $rank_i > 0$; and r_i is 0 if no answer is found, and M is total number of test cases. The $rank_i$ is the rank of the first right answer of the i^{th} test case. That is, if the first right answer is rank 1, the score is 1/1; if it is at rank 2, the score is 1/2, and so on. The value of MRR is between 0 and 1. The inverse of MRR denotes the average position of the correct answer in the proposed candidate list. The higher the MRR is, the better the performance is.

Table 1. Performance of models wo/with error recovery

Models	Recall	MRR
ASR only	0.79	0.33
ASR + Web	0.94	0.94
ASR/Pre-Removed + Web	0.59	0.48

Table 2. Distribution before/after error recovery

	Before Error Recovery							After Error Recovery						
Length of NEs	Number of Matching Characters							Number of Matching Characters						
	0	1	2	3	4	5	6	0	1	2	3	4	5	6
2	11	23	0	-	-	-	-	13	21	0	-	-	-	-
3	6	29	76	0	-	-	-	6	39	64	2	-	-	-
4	6	25	90	184	0	-	-	19	52	66	62	106	-	-
5	9	10	12	77	193	0	-	11	23	36	41	53	137	-
6	0	0	1	8	20	39	0	0	3	19	12	7	5	22

Table 1 summarizes the experimental results of models without/with error recovery. In "ASR only" model, top 10 ASR strings produced at Stage 2 are regarded as answers. This model does not employ error recovery procedure. The recall rate is 0.79 and the MRR is 0.33. That is, total 79 of 100 transliteration names are recognized correctly and they appear in the first 3.03 (=1/0.33) positions. In contrast, "ASR + Web" model utilizes error recovery procedure. PAT candidates extracted from the Web are selected at Stage 4. The recall rate is 0.94 and the MRR is 0.94. Total 94 transliteration names are recognized correctly, and they appear in the first 1.06 (=1/0.94) positions on the average. In other words, when they are recognized correctly, they are always the top 1. Compared to the first model, the recall rate is increased 18.99%. In the third model, i.e., "ASR/Pre-Removed + Web" model, we try to evaluate the extreme power of error recovery. The correct transliteration names appearing in the set of ASR strings are removed before being submitted to search engine. That is, all the ASR strings submitted to search engine contain at least one wrong character. In such cases, the recall rate is 0.59 and the MRR is 0.48. That means 59 transliteration names are recovered, and they appear in the first 2.08 (=1/0.48) positions on the average. We further examine the number of errors in "ASR/Pre-Removed + Web" model to study the error tolerance of using the Web. Table 2 shows the analyses from the length of transliteration names (row part), and the number of matching characters (column part). For a transliteration name of length l, $0 \leq$ the number of matching characters $\leq l$. Each cell denotes how many strings belong to the specific category. For example, before error recovery, there are 6, 25, 90, 184, and 0 strings of length 4, which have 0, 1, 2, 3, and 4 matching characters with the corresponding answer keys, respectively. After error recovery, there are 19, 52, 66, 62, and 106 strings of length 4, which have 0, 1, 2, 3, and 4 matching characters with the answer keys, respectively. In other words, the recovery procedure corrects some wrong characters. The number of 1-character (2-character) errors is decreased from 184 (90) cases to 62 (66) cases, and total correct strings are increased from 0 to 106.

Table 3 shows the effects of the error positions (row part) and the string lengths (column part). Here only the cases of single errors are discussed. The cell denotes how many strings are recovered under the specific position and length. For example, total 37, 35, 20, and 17 single errors for strings of length 4 appearing at positions 1, 2, 3, and 4, respectively, can be recovered by the Web. From the length issue, the longer strings

Table 3. Effects of error positions and string lengths

Error Positions	Length=2	Length=3	Length=4	Length=5	Length=6	Total
Position 1	0	0	37	42	7	86
Position 2	0	2	35	42	4	83
Position 3	-	0	20	19	9	48
Position 4	-	-	17	24	3	44
Position 5	-	-	-	14	3	17
Position 6	-	-	-	-	1	1
Total	0	2	109	141	27	279

have better recovery capability than the shorter strings. In the experiments, 0% (=0/34), 1.80% (=2/111), 35.74% (=109/305), 46.84% (=141/301), and 39.71% (=27/68) of strings of lengths 2, 3, 4, 5, and 6 can be recovered, respectively. From the position issue, the errors appearing in the beginning are easier to be recovered than those appearing in the end. The experiments show that 30.82% (=86/279), 29.75% (=83/279), 17.20% (=48/279), 15.77% (=44/279), 6.09% (=17/279), and 0.36% (=1/279) of strings with wrong character appearing at positions 1, 2, 3, 4, 5 and 6 can be recovered, respectively.

4.2 Performance of speech recognition task

The set of 100 transliteration names in Section 4.1 are spoken by 2 males and 1 female, so that 300 transliteration names are recorded. We employ HTK and SRILM to get the best 100 syllable lattices (N Best, N=100). TCC-300 dataset for Mandarin is used to train the acoustic models. There are 417 HMM models and each has 39 feature vectors. The syllable accuracy is computed as: (M-I-D-S)/M * 100%, where M is the number of correct syllables, I, D, and S denote the number of insertion, deletion and substitution errors. The syllable accuracy is 76.57%. Easily, for estimating character recovery ability, we consider the exactly correct character number. The accuracy, ASR only and ASR+Web string character errors, are computed as:

$$\sum_{i=0}^{T} \max_{j=1 to K} \left(\frac{Sim(AnsKey_{ij}, ASR_i)}{Word_{Length}(TestName_i)} \right) \quad (2)$$

and

$$\sum_{i=0}^{T} \max_{j=1 to K} \left(\frac{Sim(AnsKey_{ij}, PAT_i)}{Word_{Length}(TestName_i)} \right) \quad (3)$$

,where T is total test number and K is answer keys number with a test name i. Table 4 shows character level results. The “ASR+Web” model has 3.82% performance increasing to the “ASR Only” model on the average. Table 5 shows word level results. The error recovery mechanism supported by the “ASR+Web” model improves the recall rate and the MRR of the “ASR Only” model from 0.20 and 0.054 to 0.37 and 0.290,

respectively. In other words, the average ranks of the correct transliteration names are moved from the 18th position (=1/0.054) to the 3rd position (=1/0.290) after error recovery.

5 Conclusions

This paper employs the web corpus to correct transliteration name recognition errors. Web fuzzy search proposes useful patterns for error recovery. Fault tolerance experiments show that longer transliteration names have stronger tolerance than shorter transliteration names, and the wrong characters appearing in the beginning of a transliteration name are relatively easier to be corrected than those appearing in the end. Thus, the improvement of character level accuracy will be helpful to the recovery mechanism, and vice versa. The ASR model integrated with the recovery mechanism by the Web search facilitates the spoken access to the Web directly.

Table 4. Performance on character level

ASR Only (Character Level Accuracy)					ASR + Web (Character Level Accuracy)				
Top 1	Top 2	Top 3	Top 4	Top 5	Top 1	Top 2	Top 3	Top 4	Top 5
38.74%	43.58%	46.97%	48.75%	50.04%	43.18%	47.78%	49.96%	52.30%	53.91%

Table 5. Performance on word level

ASR Only (Word Level)		ASR + Web (Word Level)	
Recall	MRR	Recall	MRR
0.20	0.054	0.37	0.290

Acknowledgements

Research of this paper was partially supported by National Science Council, Taiwan, under the contracts NSC93-2752-E-001-001-PAE and NSC94-2752-E-001-001-PAE

6 References

[1] MUC *Message Understanding Competition*, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html, 1998.

[2] Lin, W.-C., Lin, M.-S. and Chen, H.-H., "Cross-Language Image Retrieval via Spoken Query", Proc. of RIAO 2004, 524-536, 2004.

[3] Appelt, D. E. and Martin, D., "Named Entity Extraction from Speech: Approach and Results Using the TextPro System", *Proc. of DARPA Broadcast News Workshop*, 1999, 51-54, 1999.

[4] Miller, D., Boisen, S., Schwartz, R. L., Stone, R., and Weischedel, R. M. "Named Entity Extraction from Noisy Input: Speech and OCR", Proc. of 6th Applied Natural Language Processing Conference, 316-324, 2000.

[5] Huang, F. and Waibel, A., "An Adaptive Approach of Name Entity Extraction for Meeting Application", Proc. of 2002 Human Language Technology Conference, 2002.

[6] Chen, H.-H., "Spoken Cross-Language Access to Image Collection via Captions", Proc. of 8th Eurospeech, 2749-2752, 2003.

[7] Keller, F. and Lapata, M., "Using the Web to Obtain Frequencies for Unseen Bigrams", Computational Linguistics, 29(3): 459-484, 2003.

[8] Gonnet, G. H., Baeza-Yates, R. A. and Snider, T., "New Indices for Text: PAT Trees and PAT Arrays", In Information Retrieval Data Structures Algorithms, Frakes and Baeza-Yates (eds.) Prentice Hall, 66-82, 1992.

[9] Chen, H.-H., Yang, C.-H., and Lin, Y. "Learning Formulation and Transformation Rules for Multilingual Named Entities", Proc. of ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition, 1-8, 2003.

[10] Voorhees, E., "The TREC-8 Question Answering Track Evaluation", Proc. of the 8th TREC, 23-37, 1999.

[11] Lee-Feng Chien, "PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval," ACM SIGIR Forum, 31, July 1997, 50-58.

Appendix A

Transliteration name	Answer keys list	Transliteration name	Answer keys list
科羅拉多	克羅拉多 柯羅拉多 科羅拉多	喬治克隆尼	喬治克隆尼 喬治克龍尼 喬治柯隆尼
加利福尼亞	加里福尼亞 加利福尼亞 加利弗尼亞	丹佐華盛頓	丹佐華聖頓 丹佐華盛頓
喬治亞	喬治亞 喬治雅	湯姆克魯斯	湯姆克魯斯
密西根	密希根 密西根	強尼戴普	強尼戴普
阿拉斯加	阿拉斯加	湯姆漢克斯	湯姆漢克斯
北卡羅萊納	北卡羅萊納 北卡羅萊那 北卡羅來納 北卡羅來那	*芮妮齊薇格	芮妮齊薇格 芮尼齊維格
康乃狄克	康乃迪克 康乃狄克 康迺迪克	莎莉賽隆	莎莉賽隆 莎莉塞隆 沙莉賽隆
德拉瓦	德拉瓦	妮可基嫚	妮可基嫚 尼可基曼 妮可基曼
佛羅里達	佛羅里達 佛羅理達	茱莉安摩爾	朱利安摩爾 茱莉安摩爾 朱莉安摩爾
南卡羅萊納	南卡羅萊納 南卡羅萊那 南卡羅來納 南卡羅來那	茱莉亞羅勃茲	茱莉亞羅勃茲 朱利亞羅伯茲 朱利亞羅勃茲 朱莉亞羅伯茲 茱莉亞羅伯茲
*夏威夷	夏威夷 夏威宜	威爾史密斯	威爾史密斯
愛荷華	艾荷華 愛何華 愛荷華	維果莫天森	維果莫天森 維果摩天森 維果墨天森
愛達荷	艾達荷 愛達荷	麥特戴蒙	麥特戴蒙
伊利諾	伊利諾 伊立諾 依利諾	休傑克曼	休傑克曼 休杰克曼
*印地安那	印地安那 印地安納 印弟安納	托貝馬奎爾	托貝馬奎爾
堪薩斯	坎薩斯 堪薩斯	鄔瑪舒嫚	烏瑪舒曼 鄔瑪舒曼
肯塔基	肯塔基	琪拉奈特莉	琪拉奈特莉 奇拉奈特莉
路易斯安那	路易斯安納	凱特貝琴薩	凱特貝琴薩
麻薩諸塞	麻薩諸塞 馬薩諸塞 麻塞諸塞	荷莉貝瑞	荷莉貝瑞 荷利貝瑞
緬因	緬因	安潔莉娜裘莉	安潔莉娜裘莉
馬里蘭	馬里蘭 馬利蘭	柴克巴爾夫	柴克巴爾夫
亞利桑那	亞利桑納 亞歷桑納 亞利桑那 亞歷桑那	布萊德彼特	布萊德比特 布萊德彼特 布萊得比特 布萊得彼特
明尼蘇達	明尼蘇達 明尼蘇答	金凱瑞	金凱瑞
密蘇里	米蘇里 密蘇里	柯林法洛	柯林法洛 科林法洛
密西西比	密西西比	裘德洛	裘德羅 裘德洛
蒙大拿	蒙大納 蒙大那 蒙大拿	娜塔莉波曼	娜塔利波曼 娜塔莉波曼

內布拉斯加	內布拉斯加	凱特溫絲蕾	凱特溫斯雷 凱特溫斯蕾 凱特溫絲雷 凱特溫絲蕾
阿拉巴馬	阿拉巴馬	*珍妮佛嘉納	珍妮佛嘉納
北達科他	北達科他 北達科塔	*茱兒芭莉摩	茱兒芭莉摩
新罕布夏	新漢布夏 新罕布夏	姚明	姚明
紐澤西	紐澤西	俠客歐尼爾	俠克歐尼爾 俠客歐尼爾
*新墨西哥	新墨西哥	凱文賈奈特	凱文加奈特 凱文賈奈特
內華達	內華達	*崔西麥格瑞迪	崔西麥格瑞迪 崔西麥葛瑞迪
紐約	紐約	柯比布萊恩	柯比布萊恩 科比布萊恩
俄亥俄	俄亥俄	文斯卡特	文斯卡特 文思卡特
奧克拉荷馬	奧克拉荷馬 奧克拉荷瑪 奧克拉河馬	提姆鄧肯	提姆鄧肯
奧勒岡	奧勒岡	葛蘭特希爾	格蘭特希爾 葛蘭特希爾
賓汐法尼亞	賓希法尼亞 賓西法尼亞 賓夕凡尼亞	勒布朗詹姆斯	勒布朗詹姆斯 勒布郎詹姆斯
*羅德島	羅德島	艾倫艾弗森	艾倫艾弗森 艾倫埃弗森 艾倫艾佛森
阿肯色	阿肯色 阿肯瑟 阿肯塞	*小歐尼爾	小歐尼爾
南達科他	南達科塔 南達科他 南達柯塔	拉希德華萊士	拉希德華萊士 拉西德華萊士
田納西	田納西 田那西	普林斯	普林斯
德克薩斯	德克薩斯 得克薩斯	賈米森	賈米森
*猶他	猶他	比盧普斯	比魯普斯 比盧普斯
佛蒙特	佛蒙特	斯托賈科維奇	斯托賈克維奇 斯托賈科維奇 斯托賈可維奇
維吉尼亞	維基尼亞 維吉尼亞 維吉尼雅	德克諾維茨基	德克諾維茨基 德克諾威茨基 德克諾維斯基
華盛頓	華聖頓 華盛頓 華勝頓	班華萊士	班華萊士 班華勒斯
西維吉尼亞	西維基尼亞 西維吉尼亞 西維吉尼雅	卡梅隆安東尼	卡梅隆安東尼 卡麥隆安東尼
威斯康辛	威斯康辛 威斯康新	斯塔德邁爾	斯塔德麥爾 斯塔德邁爾 斯塔達邁爾
懷俄明	懷俄明	基里連科	基里連科

***.** “印、島、兒、猶、小、芮” characters are not in training set and “尼、妮”, ” 辛、新” and “奇、琪” differentia of frequency is too high.