COMPUTATIONAL TECHNIQUES FOR IMPROVED NAME SEARCH

Beatrice T. Oshika
SPARTA, Inc.
2560 Ninth Street
Suite 315B
Berkeley, CA 94710

Filip Machi
Department of Mathematics
University of California, Berkeley
Berkeley, CA 94720

Bruce Evans
TRW
MS 02/1761
One Space Park
Redondo Beach, CA 90278

Janet Tom
Santa Monica Research Center
Unisys
2400 Colorado Avenue
Santa Monica, CA 94710

ABSTRACT

This paper describes enhancements made to techniques currently used to search large databases of proper names. Improvements included use of a Hidden Markov Model (HMM) statistical classifier to identify the likely linguistic provenance of a surname, and application of language-specific rules to generate plausible spelling variations of names. These two components were incorporated into a prototype front-end system driving existing name search procedures. HMM models and sets of linguistic rules were constructed for Farsi, Spanish and Vietnamese surnames and tested on a database of over 11,000 entries. Preliminary evaluation indicates improved retrieval of 20-30% as measured by number of correct items retrieved.

1.0 INTRODUCTION

This paper describes enhancements made to current name search techniques used to access large databases of proper names. The work focused on improving name search algorithms to yield better matching and retrieval performance on data-bases containing large numbers of non-European 'foreign' names. Because the linguistic mix of names in large computer-supported databases has changed due to recent immigration and other demographic factors, current name search procedures do not provide the accurate retrieval required by insurance companies, state motor vehicle bureaus, law enforcement agencies and other institutions. As the potential consequences of incorrect retrieval are so severe (e.g., loss of benefits, false arrest), it is necessary that name search techniques be improved to handle the linguistic variability reflected in current databases.

Our specific approach decomposed the name search problem into two main components:

- Language classification techniques to identify the source language for a given query name, and
- Name association techniques, once a source language for a name is known, to exploit language-specific rules to generate variants of a name due to spelling variation, bad transcriptions, nicknames, and other name conventions.

A statistical classification technique based on the use of Hidden Markov Models (HMM) was used as a language discriminator. The test database contained about 11,000 names, including about 2,000 each from three target languages, Vietnamese, Farsi and Spanish, and 5,000 termed 'other' to broadly represent general European names. The decision procedures assumed a closed-world situation in which a name must be assigned to one of the four classes.

Language-specific rules in the form of context-sensitive, string rewrite rules were used to generate name variants. These were based on linguistic analysis of naming conventions, pronunciations and common misspellings for each target language.

These two components were incorporated into a front-end system driving existing name search procedures. The front-end system was implemented in the C language and runs on a VAX-11/780 and Sun 3 workstations under Unix 4.2. Preliminary tests

indicate improved retrieval (number of correct items retrieved) by as much as 20-30% over standard SOUNDEX and NYSIIS (Taft 1970) techniques.

2.0 CURRENT NAME SEARCH PROCEDURES

In current name search procedures, a search request is reduced to a canonical form which is then matched against a database of names also reduced to their canonical equivalents. All names having the same canonical form as the query name will be retrieved. The intent is that similar names (e.g., Cole, Kohl, Koll) will have identical canonical forms and dissimilar names (e.g., Cole, Smith, Jones) will have different canonical forms. Retrieval should then be insensitive to simple transformations such as spelling variants. Techniques of this type have been reviewed by Moore et al. (1977).

However, because of spelling variation in proper names, the canonical reduction algorithm may not always have the desired characteristics. Sometimes similar names are mapped to different canonical forms and dissimilar names mapped to the same forms. This is especially true when 'foreign' or non-European names are included in the database, because the canonical reduction techniques such as SOUNDEX and NYSIIS are very language-specific and based largely on Western European names. For example, one of the SOUNDEX reduction rules assumes that the characteristic shape of a name is embodied in its consonants and therefore the rule deletes most of the vowels. Although reasonable for English and certain other languages, this rule is less applicable to Chinese surnames which may be distinguished only by vowel (e.g., Li, Lee, Lu).

In large databases with diverse sources of names, other name conventions may also need to be handled, such as the use of both matronymic and patronymic in Spanish (e.g., Maria Hernandez Garcia) or the inverted order of Chinese names (e.g., Li-Fang-Kuei, where Li is the surname).

3.0 LANGUAGE CLASSIFICATION

As mentioned in section 1.0, the approach taken to improve existing name search techniques was to first classify the query name as to language source and then use language-specific rewrite rules to generate plausible name variants. A statistical classifier based on Hidden Markov Models (HMM) was developed for several reasons. Similar models have been used successfully in language identification based on phonetic strings (House and Neuburg 1977, Li and Edwards 1980) and text strings (Ferguson 1980). Also, HMMs have a relatively simple structure that make them tractable, both analytically and computationally, and effective procedures already exist for deriving HMMs from a purely statistical analysis of representative text.

HMMs are useful in language classification because they provide a means of assigning a probability distribution to words or names in a specific language. In particular, given an HMM, the probability that a given word would be generated by that model can be computed. Therefore, the decision procedure used in this project is to compute that probability for a given name against each of the language models, and to select as the source language that language whose model is most likely to generate the name.

3.1 EXAMPLE OF HMM MODELING TEXT

The following example illustrates how HMMs can be used to capture important information about language data. Table 1 contains training data representing sample text strings in a language corpus. Three different HMMs of two, four and six states, were built from these data and are shown in Tables 2-4, respectively. (The symbol CR in the tables corresponds to the blank space between words and is used as a word delimiter.)

These HMMs can also be represented graphically, as shown in Figures 1-3. The numbered circles correspond to states; the arrows represent state transitions with non-zero probability and are labeled with the transition probability. The boxes contain the probability distribution of the output symbols produced when the model is in the state to which the box is connected. The process of generating the output sequence of a model can then be seen as a random traversal of the graph according to the probability weights on the arrows, with an output symbol generated randomly each time a state is visited, according to the output distribution associated with that state.

For example, in the two-state model shown in Table 2 (and graphically in Figure 1), letter (non-delimiter) symbols can be produced only in state two, and the output probability distribution for this state is simply the relative frequency with which each letter appears in the training data. That is, in the training data in Table 1 there are 15 letter symbols:

Table 1. Sample Training Data

Training Data for Example
a
ab
abc
abcd
abcde

Table 2. Two State HMM Based on Sample Data

Final Hidden Markov Model Parameters Two State, State Output Model		
Output Probabilities		
Symbol	State	
	1	2
CR	1	0
a	0	0.333
b	0	0.267
c	0	0.2
d	0	0.133
e	0	0.0667
State Transition Probabilities		
From	To	
	1	2
1	0	1
2	0.333	0.667

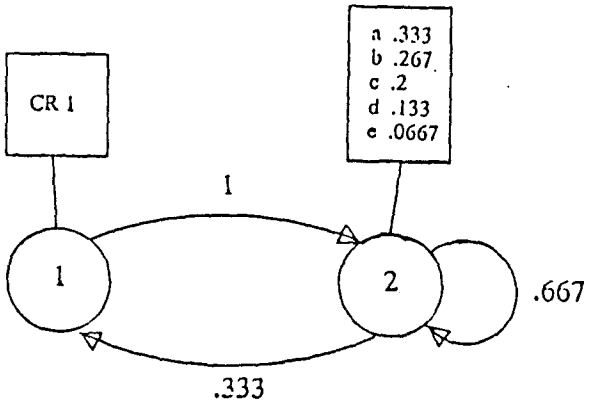

Figure 1. Graphic Representation of Two State HMM for Sample Data

Table 3. Four State HMM Based on Sample Data

Final Hidden Markov Model Parameters Four State, State Output Model				
Output Probabilities				
Symbol	State			
	1	2	3	4
CR	1	0	0	0
a	0	1	0	0
b	0	0	0	1
c	0	0	0.5	0
d	0	0	0.333	0
e	0	0	0.167	0
State Transition Probabilities				
From	To			
	1	2	3	4
1	0	1	0	0
2	0.2	0	0	0.8
3	0.5	0	0.5	0
4	0.25	0	0.75	0

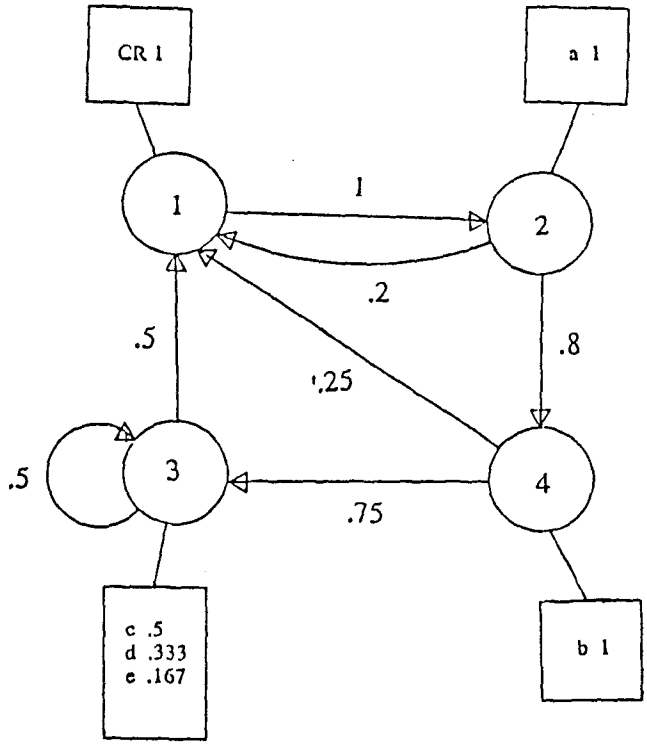

Figure 2. Graphic Representation of Four State HMM for Sample Data

Table 4. Six State HMM Based on Sample Data

Hidden Markov Model Parameters Six State, State Output Model						
Output Probabilities						
Symbol	State					
	1	2	3	4	5	6
CR	1	0	0	0	0	0
a	0	0	1	0	0	0
b	0	0	0	0	1	0
c	0	0	0	1	0	0
d	0	1	0	0	0	0
e	0	0	0	0	0	1
State Transition Probabilities						
From	To					
	1	2	3	4	5	6
1	0	0	1	0	0	0
2	0.5	0	0	0	0	0.5
3	0.2	0	0	0	0.8	0
4	0.333	0.667	0	0	0	0
5	0.25	0	0	0.75	0	0
6	1	0	0	0	0	0

Table 5. Output from Two, Four and Six State HMM for Sample Data

Outputs from Hidden Markov Models		
Two States	Four States	Six States
aadcc	ab	abcde
be	ab	abc
abcacaa	abcc	abcd
dcacc	abd	abcde
aaedb	abd	a
c	ab	abcde
caea	abc	abc
c	ab	abcd
cbc	ab	ab
ec	ab	abc
b	abc	ab
cbbcbcaebd	abcd	abc
a	a	a
ca	ab	abcd
b	abc	abcd
cb	abccdcc	abc
cde	abcc	ab
bccbabebd	ab	abc
bc	ab	abcd
dd	ab	abcd
dca	abe	abcde
ad	abed	a
e	ab	abcde
c	abe	abcd
ba	ab	ab
baea	abc	ab
b	abe	ab
ba	a	abcde
cabbd	ab	a
b	ab	a
ac	abe	ab

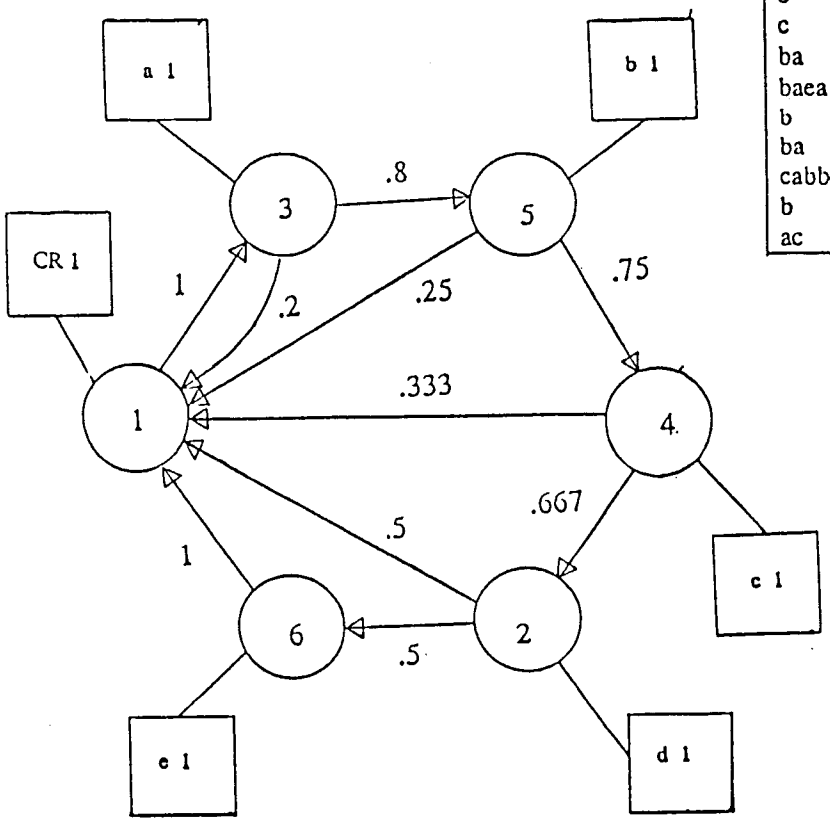

Figure 3. Graphic Representation of Six State HMM for Sample Data

five "a", four "b", three "c", etc., and the model assigns a probability of 5/15 = 0.333 to "a", 4/15 = 0.267 to "b", and so on. Similarly, the state transition probabilities for state two reflect the relative frequency with which letters follow letters and word delimiters follow letters. These parameters are derived strictly from an iterative automatic procedure and do not reflect human analysis of the data.

In the four state model shown in Table 3 (and Figure 2), it is possible to model the training data with more detail, and the iterations converge to a model with the two most frequently occuring symbols, "a" and "b", assigned to unique states (states two and four, respectively) and the remaining letters aggregated in state three. State one contains the word delimiter and transitions from state one occur only to state two, reflecting the fact that "a" is always word-initial in the training data.

In the six state model shown in Table 4 (and Figure 3), the training data is modeled exactly. Each state corresponds to exactly one output symbol (a letter or word delimiter). For each state, transitions occur only to the state corresponding to the next allowable letter or to the word delimiter.

The outputs generated by these three models are shown in Table 5. The six state model can be used to model the training data exactly, and in general, the faithfulness with which the training data are represented increases with the number of states.

3.2 HMM MODEL OF SPANISH NAMES

The simple example in the preceding section illustrates the connection between model parameters and training data. It is more difficult to interpret models derived from more complex data such as natural language text, but it is possible to provide intuitive interpretations to the states in such models.

Table 6 shows an eight state HMM derived from Spanish surnames. State transition probabilities are shown at the bottom of the table, and it can be seen that the transition probability from state eight to state one (word delimiter) is greater than .95. That is, state eight can be considered to represent a "word final" state. The top part of the table shows that the highest output probabilities for state eight are assigned to the letters "a,o,s,z", correctly reflecting the fact that these letters commonly occur word final in Spanish Garcia, Murillo, Fuentes, Diaz. This HMM also "discovers" linguistic categories, such as the class of non-word-final vowels represented by state seven with the highest output probabilities assigned to the vowels "a,e,i,o,u".

3.3 LANGUAGE CLASSIFICATION

In order to use HMMs for language classification, it was first necessary to construct a model for each language category based on a representative sample. A maximum likelihood (ML) estimation technique was used because it leads to a relatively simple method for iteratively generating a sequence of successively better models for a given set of words. HMMs of four, six and eight states were generated for each of the language categories, and an eight state HMM was selected for the final configuration of the classifier. Higher dimensional models were not evaluated because the eight state model performed well enough for the application. With combined training and test data, language classification accuracy was 98% for Vietnamese, 96% for Farsi, 91% for Spanish, and 88% for Other. With training data separate from test data, language classification accuracy was 96% for Vietnamese, 90% for Farsi, 89% for Spanish, and 87% for Other. The language classification results are shown in Tables 7 and 8.

4.0 LINGUISTIC RULE COMPONENT

For each of the three language groups, Vietnamese, Farsi and Spanish, a set of linguistic rules could be applied using a general rule interpreter. The rules were developed after studying naming conventions and common transcription variations and also after performing protocol analyses to see how native English speakers (mis)spelled names pronounced by native Vietnamese (and Farsi and Spanish) speakers and (mis)pronounced by other English speakers. Naming conventions included word order (e.g., surnames coming first, or parents' surnames both used); common transcription variations included Romanization issues (e.g., Farsi character that is written as either 'v' or 'w').

The general form of the rules is

lhs --> rhs / leftContext__rightContext

where the left-hand-side (lhs) is a character string and the right-hand-side is a string with a possible

Table 6. Eight State HMM for Spanish

Hidden Markov Model Parameters Eight State, State Output Model for Spanish								
Output Probabilities								
Symbol	State							
	1	2	3	4	5	6	7	8
CR	1	0	0	0	0	0	0	0
-	0	0	0.00427	0	0	0	0	0
a	0	0.0479	0.0133	0	0.0042	0.0753	0.324	0.219
b	0	0.00208	0	0.0681	0.00158	0.0427	0	0
c	0	0.0193	0	0.127	0.00222	0.0864	0	0
d	0	0.0755	0.0207	0.0601	0.229	0.0408	0	0
e	0	0.567	0.032	0.00169	0.00477	0.00368	0.196	0.0268
f	0	0	0	0.00875	0	0.0612	0	0
g	0	0.0207	0	0.174	0	0.052	0	0.00161
h	0	0	0	0	0	0.0825	0.0109	0
i	0	0.00432	0.0495	0	0.013	0.00193	0.164	0.00442
j	0	0.0104	0	0.0233	0	0.00295	0	0
k	0	0.00252	0	0	0	0.00123	0	0
l	0	0.0048	0.189	0.066	0.0626	0.0565	0.00559	0.0118
m	0	0.00484	0	0.118	0.00448	0.0917	0	0
n	0	0.0743	0.262	0.0697	0.0593	0	0	0.0252
o	0	0.00784	0.00968	0	0	0.0122	0.186	0.189
p	0	0.0121	0.00825	0.0132	0.0138	0.122	0	0
q	0	0	0	0.0149	0.0199	0.00551	0	0
r	0	0.0528	0.346	0.0794	0.273	0.141	0.0129	0.00279
s	0	0.0393	0.0442	0.00992	0.00899	0.0872	0	0.123
t	0	0.0339	0	0.0726	0.155	0.00288	0	0.0131
u	0	0.00162	0.00476	0	0	0	0.1	0.00671
v	0	0.015	0	0.0884	0	0.0177	0	0
w	0	0	0	0.00103	0	0.00213	0	0
x	0	0	0	0	0	0	0	0.00183
y	0	0.00198	0.013	0.0031	0.00465	0.00149	0	0.00534
z	0	0.00175	0.00287	0	0.14	0.00727	0	0.368
State Transition Probabilities								
From	To							
	1	2	3	4	5	6	7	8
1	0	0	0	0.339	0.00323	0.602	0.0548	0
2	0.00968	0.075	0.00561	0	0.0869	0.00212	0.00665	0.814
3	0.0615	0.269	0.0353	0.259	0.235	0.0097	0.0253	0.104
4	0	0.0101	0.0132	0	0.00503	0.0245	0.929	0.0182
5	0.0117	0.228	0.00477	0.00466	0.0537	0.00145	0.542	0.154
6	0	0	0.0587	0.0341	0	0.0564	0.85	0
7	0.0165	0.13	0.506	0.162	0.0627	0.00977	0.0207	0.0915
8	0.954	0	0.00169	0	0.00723	0.00216	0.00858	0.0256

Table 7. Language Classification Performance for Training Data

Language Classification Accuracy Statistics for Training Data Eight State, State Transition Output Model					
Language	Percent Classified as:				Error Rate
	Farsi	Spanish	Vietnamese	Other	
Farsi	95.5	1.4	0.1	3.0	4.5
Spanish	2.0	91.1	0.1	6.9	8.9
Vietnamese	0.3	1.0	97.8	0.9	2.2
Other	5.4	5.8	0.4	88.4	11.6

Table 8. Language Classification Performance for Test Data

Language Classification Accuracy Statistics for Non-Training Data Eight State, State Transition Output Model					
Language	Percent Classified as:				Error Rate
	Farsi	Spanish	Vietnamese	Other	
Farsi	90.1	2.7	1.0	7.1	9.9
Spanish	2.6	88.8	0.1	8.5	11.9
Vietnamese	0.6	1.6	96.0	1.8	4.0
Other	6.3	6.1	0.4	87.3	12.4

Table 9. Examples of Linguistic Rules

Rule	English Paraphrase	Examples		
		input	output	N/A
PH -> F	PH goes to F (everywhere)	Phred Stephen	Fred Stefen	...
C -> K / _A	C goes to K when it precedes A.	Cathy	Kathy	Colin
J -> J | H | G / #_	J goes to J, H or G when it is word initial.	Jimenez	Jimenez, Himenez, Gimenez	Borjas
Y -> Y | I / ._	Y goes to Y or I when it is not word initial.	Bryan Sherry	Bryan, Brian Sherry, Sherri	Yonkers
F -> F | V / _.	F goes to F or V when it is not word final.	Filip Stefan	Filip, Vilip Stefan, Stevan	Josef
C -> C | S / _[EI]	C goes to C or S when it precedes E or I.	Cespedes Garcia	Cespedes, Sespedes Garcia, Garsia	Carrillo
H ->H | J/[^CS]_[AEIOU]	H goes to H or J when it follows a letter other than C or S, and precedes A,E,I,O, or U.	Truhillo	Truhillo Trujillo	Chacon Sherri
T -> T | D / #_[^R]	T goes to T or D when it is word initial and precedes a letter other than R.	Tao Tuyet	Tao, Dao Tuyet, Duyet	Tran Kiet
IE -> IE | I | Y / _#	IE goes to IE, I or Y when it is word final	Vinnie	Vinnie, Vinni, Vinny	Pierson Mier
O -> O | E | U / S_N#	O goes to O, E or U when it follows S and precedes final N.	Anderson	Anderson,Andersen, Andersun	Andersons Anderzon

weight, so that the rules could be associated with a plausibility factor.

Rules may include a specific context; if a specific environment is not described, the rule applies in all cases. Table 9 shows sample rules and examples of output strings generated by applying the rules. The 'N/A' column gives examples of name strings for which a rule does not apply because the specified context is absent. An example with plausibility weights is also shown.

5.0 PERFORMANCE

Although the statistical model building is computationally intensive and time-consuming (several hours), the actual classification procedure is very efficient. The average cpu time to classify a query name was under 200 msec on a VAX-11/780. The rule component that generates spelling variants can process 100 query names in about 2-6 cpu seconds, the difference in time depending on average length of name.

As for retrieval performance, in a test of 160 query names (including names known to be in the database and spelling variants not known to be in the database), there were 111 hits (69%) using NYSIIS procedures alone and 141 hits (88%) using the front-end language classifier and linguistic rules and sending the expanded query set to NYSIIS.

In recent work, this technique has been extended to include modeling a database of Slavic surnames. Language classification accuracy based on a combined database of 13000 surnames representing Spanish, Farsi, Vietnamese, Slavic and 'other' names, with combined training data (1000 names from each language group to build each language model) and test data (remaining 8000 names), is 96.8% for Vietnamese, 87.7% for Farsi, 86.9% for Spanish, 86.5% for Slavic, and 82.9% for 'other'.

6.0 REFERENCES

Ferguson, John D., Ed. 1980 Symposium on the Application of Hidden Markov Models to Text and Speech, Institute for Defense Analyses, Communications Research Division, Princeton, New Jersey.

House, Arthur H. and Neuburg, Edward P. 1977 Toward Automatic Identification of the Language of an Utterance, Journal of the Acoustical Society of America, 62 (3):708-713.

Li, K. P. and Edwards, Thomas J. 1980 Statistical Models for Automatic Language Identification, Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Denver, Colorado, 884-887.

Moore, Gwendolyn B.; Kuhns, John L.; Trefftzs, Jeffrey L.; and Montgomery, Christine A. 1977 Accessing Individual Records from Personal Data Files Using Non-Unique Identifiers. Computer Science and Technology, National Bureau of Standards Special Publication 500-2, Washington, D.C.

Taft, Robert L. 1970 Name Search Techniques. New York State Identification and Intelligence System, Special Report No. 1, Albany, New York.