Computer Recognition of Sanskrit-Based Indian Names

Anurag Srivastava and V. Rajaraman

Abstract—**An interesting problem relating to cognition oriented task of humans is the *name recognition* problem. Humans are so apt in performing this task that they hardly realise the difficulties involved in it. Formally, name recognition means identifying the native origin of people from their names. A subproblem of this task is to classify whether a name belongs to one's own country or not. The authors analyze and then formalize the procedure followed by humans to perform this task. An expert system for recognizing names of persons of Indian origin is then described. The system is based on the phonetic information contained in the name, which in turn depends upon the language of the corresponding country (in our case, Sanskrit).**

I. Introduction

Application of expert systems and knowledge based systems in experience oriented and classification tasks has been widely established now [1], [2]. A major set of cognition oriented problems like vision, speech recognition, face recognition etc. still pose an open challenge to the researchers working in the area of artificial intelligence (AI) applications. One such interesting but ***hard*** problem is of recognizing from a name one's country of origin. All of us do this task very easily while it is difficult for a computer to perform the same task. By ***name recognition*** we mean ***guessing*** about the country of origin of a person from one's name. For example, Kowalski seems to be a Polish name, Huang Ching a Chinese name, Vostodorsky a Russian name etc. It can be observed that most people accept the above classification. How did we reason in deciding on this classification? We urge the reader to judge the hardness of the task before we discuss our approach to solve the problem.

In this paper we describe an expert recognition system that will recognize names of persons of Indian origin from among a large set of names. This recognition system is used to create a data base of persons of Indian origin working all over the world in the area of biotechnology.

We begin with the discussion of various possible solution strategies that can be used to solve the problem. Section II describes the exact methodology adopted to build this expert system. Section III discusses the uncertainty issues encountered and the performance evaluation, and Section IV the overview and a learning model of the name recognizer. Section V gives the results.

The problem of name recognition is semantic as well as syntax oriented. Thus the solution to be adopted should be such that it analyses the given character string (name) not just by some syntactic techniques but also uses the underlying semantics of name formation. In this section we will explore the various possible ways of solving this problem and their relative merits and demerits.

Manuscript received July 28, 1989; revised December 2, 1989 and June 18, 1990. This work was supported by the Department of Biotechnology of the Government of India.

The authors are with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore *560* 012 India.

IEEE Log Number 9039990.

A. Exhaustive Search

One brute-force method of classifying a name as belonging to a country would be to create an exhaustive data base of names of that country (may be from a voter's list) and then use a hashed search of the given name in this set. This brute-force method of solving the problem obviously lacks intelligence and is merely syntactic. It does not use any semantic knowledge about name, construction. In addition it is observed that we do not always classify a particular name only on the basis of whether we know it or not.

B. Syntactic Pattern Recognition Approach

Another syntactic approach for solving this problem is the syntactic pattern recognition approach [3]. That a considerable number of names are derived from a root word is the basis for this approach. This root word can be the name of a God in a Pantheon, e.g. Rama, Shiva etc. of the Hindu pantheon or a saint's name, e.g. John, Martin, etc. or some specific word with an important meaning associated with it. The syntactic PR approach looks for such regularities and tries to generate the underlying grammar associated with the samples.

To elaborate further, consider the case of Indian names. A subset of Indian names possess such regularity. If we can derive a grammar that can parse all Indian names then we have obtained the solution. The following grammar has been proposed *to* parse a subset of Sanskrit-based Indian names.

⟨Indian-Name⟩ := (Prefix)(Middle) (Suffix)
(Prefix) := Tri|Su|An|. . .|⟨empty⟩
(Suffix) := An|Ani|Am|Kar|. . .|⟨empty⟩
(Middle) := (God name)|⟨God name⟩(Middle)
(God name) := Ram|Shiva|Gopal|Krishna|Murthy|. . .

Performance of this method will be inferior as compared to the previous approach, as a very small subset of Indian names may follow such a grammar.

C. *Rule-Based Expert System*

Rule-based systems have been built to succesfully solve problems of this kind where uncertainty and human experience play an important role [1], [2]. In all such problems no definite way of finding a solution exists and hence one has to look for some heuristics. This approach is nearer to the human way of solving some recognition problems. At this point our concern is to understand the human way of decision making and then use that information in solving the same problem with the help of a computer.

*1) **Specific Rules Identification:*** All of us would have seen people guessing about one's native state/country from one's name. For example, persons with the names Kathkar, Vengsarkar, Madhukar etc., are usually from Maharashtra, a state in India. Similarly Bolshyolov, Michalski, Yomomoto, etc., are identified as Russian, Polish, and Japanese respectively. From the examples it can be deduced that the ending of the name plays an important role in this decision-making process. Thus one of the rules could be to use the ending information. The presence of root words can be taken as one of the rules. These rules are useful but only as side-tools. The main reasoning procedure i.e., the governing rule is not based on such narrow observations.

*2) **Structural or Semantic Rules Identification:*** The most important observation we make in this paper is that when we read a name, we pronounce it in our mind and then manipulate on the phonetic information present in the name. We associate the pronounciation of the given name with our phonetic knowledge base. If the syllables in the name produce phonemes which match with the phonemes of our native languages, then we classify that name as that of our country.

0018-9472/91/0100-0287$01.00

To conclude, we can say that the process of name recognition is done in the following three major phases.

1) A text to speech conversion phase.
2) Phoneme matching or association phase.
3) Additional rule usage to improve recognition.

This basic idea of phonetic structure analysis of name is the governing rule to which we add the other previously mentioned rules to arrive at our decision.

II. Phonetic-Based Name Recognizer

In this section we will describe in detail the theoretical foundations and methods used in text to speech conversion phase for name recognition identified in the last section. We also describe the actual implementation of the recognizer.

A. Theoretical Foundations

From the type of problem that we are going to handle and the phases identified in the last section it can be inferred that the names have a relation with the languages spoken in a given country. In fact the names are often a subset of the vocabulary of the country's language. When names are written in English, then the mapping is from the corresponding country's language phonetic system to that of English, in our case from Indian languages to English. However, the problem here is the reverse, i.e, we are given the name written in English and we want to identify the phonetic structure of an Indian language in the given name.

This mapping between English and any other language plays an important role in the text to speech conversion phase owing to different sizes of the alphabet that leads to ambiguities in representing the phonemes of that language when the name is written in English. As most of the Indian languages have their origin in Sanskrit, we describe this mapping for Sanskrit and also formalize the previously identified phases by determining the structural information about name formation. This phase is domain dependent, i.e., on that country's language for which such a system is to be built. Now we describe the Indian name structure determination phase based on Sanskrit phonemes.

Sanskrit has 32 usable consonants and 13 vowels. English has five vowels and 21 consonants and hence there does not exist a bijection between Sanskrit and English alphabets.

This leads to three types of problems. 1) Two or more than two letters of English are used to represent one letter of Sanskrit. 2) One letter of Sanskrit may have more than one representation in English. This is common for vowels. 3) Some letters do not exist at all in English and hence two different Sanskrit letters may have the same representation in English.

As a result of these representational problems, the associated text to speech conversion process leads to ambiguities in the exact pronunciation of a name and hence may result in misclassification.

The order in which the letters appear in a name affects name recognition. This ordering restriction is imposed by the sound associated with some syllables not being present and not being used often in Indian languages. We will now try to discover the set of rules which constrain the ordering of letters. Before going further we introduce a new term called the valid *syllabophonic* (SP) *unit*. A valid SP unit is the minimal length string of phonetically combinable letters of a language. These units forms the basic structural unit in the noun construction in Sanskrit. A valid SP unit refers to those units in which the appearance of consonants (vyanjanas) and vowels (svaras) is such that they are found in spoken language and are used often in the words present in the dictionary of Indian languages. The decision as to which are usable units is a matter of experience and an individual's own language's knowledge.

An SP unit can be elongated due to the presence of a vowel or a nasal or a consonant, which is compatible with the last letter in the unit.[1] If one observes the structure of an Indian name, it can be observed that the name consists of one or more such valid SP units or their elongations. An interesting property of these valid SP units is that when they join to form a larger word a distinct pause comes at the boundary of the two units when the word is pronounced. For example, while pronouncing the name Ramlal there is no tendency to vocalize it as "raml" and "al," a natural pause is present at the boundary of "ram" and "lal." This observation led us to the idea of introducing such a unit.

B. Phonetic Recognizer's Design Philosophy

Based on the discussions of the previous subsection the three phases in name recognition described in Section I can be reformulated as the following three steps.

1) Identification of syllabophonic units or their elongations in the name.
2) Association:- i.e. matching the SP units in the name obtained in step 1 with the valid set of SP units in the knowledge base.
3) Application of empirical rules to enhance recognition performance.

It was found that two consonants and a vowel in between, generally are sufficient to produce a single minimal such unit.[2]

Using the above structural definition of an SP unit and the commonality of it's usage in the language, a set of valid SP units which form the factual knowledge base for this expert system was constructed. Around 550 such units have been obtained. The SP units present in the name are matched with this knowledge base. Based on how many SP units match with the knowledge base of valid SP units the decision to classify a name is taken. We describe step 1 of how the recognition process.

SP Unit Identification: The method adopted to perform this step was guided by the phonetic structure of Sanskrit. Following rules were used in the recogniser to obtain the parts of a name.

1) As a SP unit consist of a consonant, it is observable that whenever two consonants are adjacent there is a likelihood that a break up can be performed.
2) The property of some consonants to go together is termed as compatibility and is due to Indian languages allowing half consonant concatenated with another consonant as a single letter for some set of consonants. If the two consonants are compatible then they go together and no division will be done at that point, otherwise this point produces one SP unit or its elongation. The procedure is again applied on the rest of the name.
3) As a result of above step it may happen that one does not find any place of division in a name *e.g.* Badrinath ("d" and "r" are compatible in the context of "a") using the first rule. The second part uses the definition of SP unit and its elongation to obtain the parts of a name. The rules comprising these exceptions were also determined by experimenting with this part of the recognizer on a set of Indian names. This procedure along with the step 2 thus acts as the heart of the recognizer.

[1]This is done to accomodate the different styles of transliteration for some pronunciation in various parts of the country *e.g.* (Ra:jðsʊ mðn) is spelt Rajsuman in north and Rajasuman in south India.

[2]Sanskrit and most Indian languages have consonants that we call *joint consonants*, e.g., 'jh', 'th'. They belong to type a) problem specified earlier. Whenever a consonant is being included in a SP unit the joint consonants are taken care of and hence whether the name is Latharaman or Lataraman the breakup is Lat/th(a)+Ram(an).

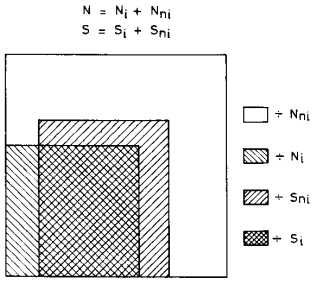

Fig. 1. Performance diagram of name recognizer. $N = N_i + N_{ni}$. $S = S_i + S_{ni}$.

The observations made in the last two subsections about the structure of Indian names provides a guideline to tackle this problem for foreign countries' names as well. The structural properties of a name are highly influenced by the underlying phonetic structure of the corresponding language. Japanese names were found to follow a similar trend as Indian names. It is conjectured that similar rules can be identified for finding the native origin of a name of many countries based on the respective languages of that country.

III. Improving Recognition Performance

A. Performance Criteria

Having found phoneme based rules in steps 1 and 2 they are applied to a set of names to find how well they perform the recognition task. We formulated the following performance criteria:

Given a set of names:

a) How many Sanskrit-based Indian names does the system miss?
b) How many non-Indian names are classified as Indians?

These two criteria can be technically expressed by the following terms [4].

a) Recall
b) Precision

To define these terms exactly consider Fig. 1.

Let N be the number of names in the universe U on which the recogniser is applied. Let N_i be the actual Sanskrit-based Indian names in the set U. Let S be the names selected by the recognizer. In these S names let S_i be the actual Sanskrit-based Indian names and S_{ni} be the incorrectly classified names. Then the performance measures are defined as follows:

1) Recall $= S_i / N_i$ and
2) Precision $= 1 - S_{ni} / S$.

It was found that the recall was 51% and precision was 55% after applying the recognition rules based on phonetic structure. The reason for missing or selecting wrong name as an Indian name is due to the reason that some of the valid SP units also appear in other countrys' names. In such a case, the names that have these valid SP units and are actually not Indian may possibly be classified as Indian. On the other hand some of the names that are Indian may have valid SP units less in number than the other SP units. Due to this some of the Indian names may be missed by the system. In order to improve the recall and precision, it was found necessary to introduce some more rules. These rules may be termed as precision enhancement rules and recall enhancement rules that take care of the above misclassifications made by the system. This constitutes step 3 of the recognition process. Following examples will enhance the understanding about why certain names were missed or wrongly selected. Ambacher is a Sanskrit-based Indian name with two part Amba and Cher. Although Amba is a valid SP unit elongation but Cher is not. Nagakoto is a non-Indian name with two parts Naga and Koto where Naga is a valid SP unit elongation of Nag but Koto is not. In step three we framed rules to take care of such cases that greatly affected the system's performance.

B. Precision Enhancement Rules

To quantify the amount of confidence in classifying a name two factors are associated with each name, called as Indianness factor (IF) and non-Indianness factor (NF).

The two factors were evaluated as follows:

IF(name) $= \Sigma$ IF(SP units in the name)

IF(SP unit) $= \Sigma$ IF(letters constituting that unit and their positions in it.)

For example,

IF(RAJ) = IF(R, 1) + IF(A, 2) + IF(J, 3)

In the same way the NF for the name can also be defined.

The confidence is then evaluated by using the two factors associated with the names as follows:[3]

Confidence factor(name) = IF(name) − NF(name).

The first or primary decision to classify the name depends on two factors

a) the number of valid SP units present in the name, and
b) the confidence factor for that name.

The name will be selected if a) the number of valid SP units is greater than or equal to the number of invalid SP units and b) the confidence factor for it is greater than a threshold value set to improve precision.

The threshold determination and the condition a) were introduced to improve precision. After the name was selected at primary level, it was then further analysed for rejection (secondary level) by a stricter and specific set of 60 rejection rules. These were based on a dictionary of SP units with a common general characteristics of pronunciation matching Japanese and European names.

C. Recall Enhancement Rules

To improve recall a set of distinct Indian name endings such as "kar," "jee," "ani," etc. were kept in a dictionary. Also a dictionary of short distinctly Indian names such as Singh, Das, Gupta etc. was created. These rules overrode rejection rules and distinctly improved recall without impairing precision.

IV. Name Recognizer's Overview

A. Program Structure

The overall structure of the recognition program developed by us is as shown in Fig. 2. The complete program is divided into the following three parts.

a) The interface unit called the extractor.
b) The rule based splitter unit called the break name.
c) The decision maker unit called the select-reject.

As the recognizer's actual application is in processing data from a large data base tape, there is no interactive and explanation part attached to it. If desired it can be added as a separate routine. The extractor is used to produce the data in the format desired by the other two parts of the program. It is used to

[3]The use of Bayesian approach by defining a single factor for an SP unit, did not seem to work because a precondition to it is that the sum of all such factors should be 1. This enforcement can be made on the individual alphabets' frequency but not on the SP unit which is made from these alphabets and forms the evidence for deciding it to be Indian or not ie., $\Sigma P(\text{e}/\text{Indian}) \neq 1.0$. More details are available in [5].

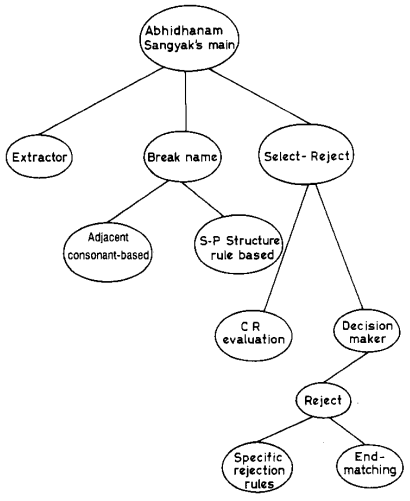

Fig. 2. Overall view of name recognizer.

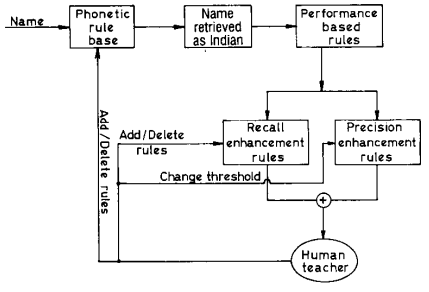

Fig. 3. Learning model of name recognizer.

identify and create the first name, last name and initials arrays from the given input file from the tape.

The break name or the splitter unit forms the major component of the recognizer. It consists of approximately 25 rules. The decision maker unit evaluates the confidence factor for a name and employs the enhancement rules to finally classify a name to be of Indian origin or not. This unit consisted of rules for rejection, rules for end-matching and a few rules for recall enhancement.

B. Actual Usage of the Name Recognizer

The recognizer is the major part of the actual system required to create the database of scientists of Indian origin working in the field of biotechnology in the world. The database should have the following information.

- Scientists Name
- Affiliation
- Major Areas of Interest.

The input file are data bases of articles published in various journals in biotechnology. These articles have besides the name of the article, the author(s)' name, address and key terms describing the article. The recognizer filters out articles published by authors of Indian origin to create the data base. The main purpose of the Indian biotechnologists' data base is to identify experts for possible consultancy assignments in India.

C. Learning Model of the Recognizer

Based on the procedure adopted to develop the recognizer, a learning model for the system can be proposed. The structure of such a learning system is shown in Fig. 3. The inner loop is a parameter base learning in which the threshold for the confidence factor is learnt. The outer loop is based on the phonetic misclassification information and leads to change in the phonetic rule base.

V. Conclusion

With the assumptions made in the design phase, the validity of which is proved only intuitively, there is a scope of improvement in the performance of the system with experience. Before the system was used in the actual application, tests were done on real data sets and the performance of the system was evaluated.

The most difficult aspect in evaluating the system during the testing phase was the recall determination. As defined in Section IV-A, it is the number of names of Indian origin missed by the system. The actual data sets were tested for about 30,000 names. The results for two such tests is given as follows

1) $N = 30{,}000$ (Total number of names in the data set)
   $S = 2767$ (Names selected as of Indian origin by the recognizer)
   $S_i = 2198$ (Actual names of Indian origin in S)
   Names missed by the system (manually determined) $= 216$
   $N_i = 2198 + 216 = 2414$
   Recall $= 2198/2414 = 0.9107$ i.e. 91.07%
   Precision $= 1 - (S - S_i)/S = S_i/S = 2198/2767 = 0.795$ i.e. $= 79.5\%$
2) $N = 32{,}000$
   $S = 2919$
   $S_i = 2393$
   Names of Indian origin missed $= 256$
   $N_i = 2393 + 256 = 2649$
   Recall $= 2393/2649 = 0.9041$ i.e., 90.41%.
   Precision $= 2393/2919 = 0.819$ i.e., 81.9%

From the above results one can see that the recall is quite high. For precision evaluation one has to go through only around 3000 names. But for recall evaluation it was necessary to go through 27,000 names which was a time consuming process. It was found almost impossible for a single person to perform this job because it requires lot of patience and concentration. The above experiment led to some interesting observations and acted as a good psychological test for comparing the performance of the Computer-based recognizer and human performance. The advantage of the computer based system was quite clear after seeing the human decision process in this case [5].

Acknowledgment

Authors would like to thank the reviewers for their critical comments on the earlier version of this paper.

References

[1] F. Hayes-Roth, D. A. Waterman, D. B. Lenat, "Building expert systems," Addison-Wesley, Reading, MA 1983.
[2] W. Clancey, "Heuristic classification," *Artificial Intell.*, vol. 27, no. 3, pp. 289–350, 1985.
[3] K. S. Fu, "Syntactic methods in pattern recognition," New York: Academic, 1974.
[4] G. Salton, "Introduction to modern information retrieval," McGraw-Hill, 1983.
[5] A. Srivastava, "Abhidhanam sangyak: An expert system to create the data base of indian scientists," ME project report, Dept. Comput. Sci. Automat., Indian Inst. Sci., Jan. 1989.