Journal of Quantitative Linguistics
2006, Volume 13, Number 1, pp. 57 – 80
DOI: 10.1080/09296170500500694

Routledge
Taylor & Francis Group

Language Identification from Small Text Samples*

Kavi Narayana Murthy and G. Bharadwaja Kumar
Department of Computer and Information Sciences, University of Hyderabad, India

ABSTRACT

There is an increasing need to deal with multi-lingual documents today. If we could segment multi-lingual documents language-wise, it would be very useful both for exploration of linguistic phenomena, such as code-switching and code mixing, and for computational processing of each segment as appropriate. Identification of language from a given small piece of text is therefore an important problem. This paper is about language identification from small text samples.

In this paper, language identification is formulated as a generic machine learning problem – a supervised classification task in which features extracted from a training corpus are used for classification.

Regression is a well established technique for modelling and analysis. Regression can also be used for classification. This paper gives a clear formulation of multiple linear regression for solving a two-class classification problem. Theoretical bases for verifying the adequacy of the model for the task and for analysing the significance of individual features is included.

The method has been applied to pair wise language identification among several major Indian languages including Hindi, Bengali, Marathi, Punjabi, Oriya, Telugu, Tamil, Malayalam and Kannada. Some of these languages belong to the Indo-Aryan family while the others come from the Dravidian family of languages. Language identification was so far a largely unexplored problem in the Indian context.

Variations within and across language families have been explored. Variations with regard to sizes of test samples have also been explored. Performance is comparable to the best published results for other languages of the world.

In most of the published work in language identification so far, bytes have been taken as the fundamental units of text. Indian scripts are primarily syllabic in nature, reflecting phonetic sound units in a more or less direct fashion. The fundamental units of writing are called aksharas. One of the unique characteristics of Indian scripts is the concept of a script grammar. The script grammar, included in this paper, defines the set of valid aksharas. We hypothesize that aksharas are the more appropriate units of text in Indian

*Address correspondence to: Kavi Narayana Murthy, Department of Computer and Information Sciences, University of Hyderabad, Hyderabad 500046, India.
E-mail: knmuh@yahoo.com

languages, not characters or bytes. Our experimental results on language identification support this claim.

INTRODUCTION

There is an increasing need to deal with multi-lingual documents today. Most of language technology applications in both the text and speech domains are, however, inherently language specific. A spell checker designed for Hindi cannot be applied directly on Marathi. It becomes necessary, therefore, to first segment documents language-wise (Constable & Simons, 2000; Muthusamy et al., 1994; Giguet, 1995a; Giguet, 1995b). Then the Hindi spell checker can be used for the Hindi parts and the Marathi spell checker applied to the Marathi parts. Language identification has been incorporated or integrated into many applications including text categorization and text retrieval (Wechsler et al., 1997; Piotrowski, 1997).

Instead of viewing language identification as a document segmentation problem, it is possible to take the somewhat simpler view of classifying a given small segment of text into one of the given set of languages. In this paper we take this latter classification view. Extensions to automatic text segmentation are conceivable.

There are a large number of languages used in India. Linguists believe that there are nearly 150 different languages. Twenty two of these languages have been given constitutional recognition and are considered major languages. Many of these are official languages of state governments and widely used by media. English, Hindi and other local languages are often mixed as a matter of policy and practice. Mixing Sanskrit and local languages in a single text is also very common. In most cases it is a case of frequent code switching but code mixing is also observed. Thus language identification is all the more relevant in the Indian context.

There are also many scripts. Interestingly, the correspondence between languages and scripts is not strictly one to one – some scripts are used for writing several languages and some languages are written in more than one script. Devanagari script is used to write Sanskrit, Hindi, Marathi, Konkani and Sindhi. Sanskrit is written in almost all the different scripts. Therefore mere script identification is not sufficient. It is important to be able to identify language irrespective of the script or font being used.

Excluding Kashmiri, Sindhi and Urdu, which are written mostly in the Perso-Arabic scripts, all the other major languages are written in 10 different scripts, all of which have evolved from the ancient Brahmi script. Indian scripts are phonetic in nature – the written form reflects the basic sound units. Since the basic sound units (phonemes) used in all the Indian languages are more or less the same, ISCII (Indian Script Code for Information Interchange; Bureau of Indian Standards, 1991), a BIS standard, has chosen to implement a common code space for all the 10 scripts. For example, the /k/ sound is encoded as 179 irrespective of the language or the script used. When the text is rendered, the appropriate script and font are used for display and printing but the encoding itself is language and script independent. Thus a plain ISCII document has no explicit indication of language. Automatically identifying language from small text samples in ISCII texts is therefore very important.

It may also be noted that switching to UNICODE will not solve the problem – UNICODE provides code spaces for scripts, not necessarily one for each language. Devanagari script is used for several languages including Sanskrit, Hindi, Marathi, Nepali and Konkani. The Bengali script is also used for Assamese. Also, it is highly desirable to make a machine learning system completely generic – exploiting explicit cues such as scripts and fonts is generally not acceptable in the machine learning community.

Despite its great relevance and need, language identification has not been explored much in the context of Indian languages. In this paper we formulate the problem of language identification as a two-class supervised learning problem using multiple linear regression (MLR). Regression is a well established technique with a strong theoretical foundation. Language identification is viewed as a generic machine learning problem, a supervised classification task in which features extracted from a training corpus are used for classification. MLR has been used to estimate the weights of features and has been shown to be very effective for identification of language in the Indian context. Theoretical bases for verifying the adequacy of the model for the task and for analysing the significance of individual features are included. The model has been applied to pair wise language identification among major Indian languages including Hindi, Bengali, Marathi, Punjabi, Oriya, Telugu, Tamil, Malayalam and Kannada. Variations within and across language families and variations with regard to sizes of test samples have been

explored. Results obtained are comparable to the best published results for other languages of the world.

The following section gives a brief survey of various approaches to language identification.

A BRIEF SURVEY OF LANGUAGE IDENTIFICATION RESEARCH

In the last decade or so, corpus-based machine learning approaches have become predominant in language engineering over the knowledge-based approaches which use explicit rules hand crafted by domain experts. Recent research on language identification has been limited almost exclusively to machine learning approaches. In machine learning approaches, a set of training data is given and the machine "learns" a general rule or builds a model for performing the intended task. A machine learning system is expected to be generic and it is understood that training is based only on the intrinsic properties of the data, as expressed through a set of "features". Extraneous indicators such as clues from scripts or fonts used, header information or explicit markup tags in the document structure cannot be used. Dictionaries or word lists, lists of affixes etc. are also generally not permitted.

Machine learning can be supervised or unsupervised. In supervised learning, a set of labelled training data is given and the machine learns a general decision rule which can be used for classification of new data items. In unsupervised learning, a set of unlabelled training data is given and the machine learns to group similar data items into clusters so that new data items can be placed into the right clusters. The number of clusters may or may not be known beforehand.

Beesley (1988) proposed a language identification program for the documents in English, Spanish, French and Portuguese languages in the year 1988. Here the basic idea is that each language uses a unique or a very characteristic alphabet, and the letters of the alphabet appear with surprisingly consistent frequencies in any statistically significant text. In addition, the frequency of occurrence of sequences of two, three, four or more letters are characteristically stable within, but diverse among different natural languages. In this study the most frequent 3-grams, 4-grams etc. were used for language identification.

Dunning (1994) proposed a language classifier for English and Spanish. His program incorporates no presuppositions other than the assumption that text can be encoded as a string of bytes. He used a Bayesian classifier to classify a given text into one of the given languages. He reported an accuracy of about 92% on 20 byte test samples and 50 K bytes of training data. The performance improved to about 99.9% when 500 bytes were examined. 5 K bytes of training text and 500 byte test data gave an accuracy of about 97%. For longer text strings (>100 bytes) and larger training sets (50 K bytes or more), more than 99% accuracy with roughly 90% confidence was reported.

Combrinck and Botha (1995) presented a text-based language identification system for 12 languages. The model was based on transition vectors, where a transition vector was either a single character or a combination of characters. A crucial part of the recognition system was the identification of the set of most distinctive, most frequently encountered sequences of characters (that is, *n*-grams) that could be associated with each language. Distinctiveness implies that the frequency of a letter combination for a given language is high relative to the frequency of occurrence in other languages. The system built a histogram of the number of hits for the various transition vectors for each language. The histogram was normalized by the total number of characters in the test set to produce a figure indicating the percentage of the test set that was found in the model of each language. A text was classified as belonging to the language model for which the score was the highest. High performance was reported on the following languages: Afrikaans, English, Sepedi, Xhosa, Zulu, Tswana, Swazi, German, Italian, French, Spanish and Portuguese.

Using an *n*-gram based algorithm, Adams and Resnik (1997) proposed a system to dynamically add language labels for whole documents and text fragments on the World Wide Web. The program used all 5-grams observed from 220 K bytes of training data. An average accuracy of 98.68% for 100 – 500 character length strings was reported for English and Spanish languages. The program gave a lower accuracy rate of 98.32% on the same training data when a trigram model was used and trigrams whose observed frequency was less than four were filtered out.

Prager (1999) proposed the Linguini system – a vector-space-based categorizer used for language identification. He used cosine similarity measure to identify the language from a given feature vector. Linguini uses dictionaries generated from features extracted from training texts,

and compares these against feature vectors generated from test inputs. Features used are character level *n*-grams, words, and combinations of the two. If only character level *n*-grams are used, it was shown that 4-grams give the best results. If full words alone are used, it was shown that words of unrestricted length did better than short words. Perhaps unsurprisingly, when dictionaries were generated by using both *n*-grams and full words as features, the best performance was obtained with the combination of 4-grams and words of unrestricted length. The average performance was 85.4% for 20-byte inputs, rising to over 99% for 130 bytes and above.

Xafopoulos et al. (2004) proposed a language identification system based on a hidden Markov model (HMM) for modelling of character sequences. This system was used to identify language in web documents automatically. English, German, French, Spanish, and Italian were considered in their experiments. They reported 99% accuracy on the test sequences of about 140 characters. Language identification has not been explored much in the context of Indian languages.

MULTIPLE LINEAR REGRESSION AS A CLASSIFICATION TECHNIQUE

In this section, we formulate the problem of language identification as a two class classification problem using MLR. We discuss the theoretical merits and practical advantages of MLR and show that MLR is simple, efficient and adequate for the problem on hand.

Regression analysis (Montgomery et al., 2001; Glantz & Slinker, 2000; Allison, 1999) is a statistical technique for investigating and modelling the relationship between variables in a system. When there are more than two variables in the system, the term multiple regression is employed. Regression is often used as a modelling technique where the value of one of the selected variables, called the response variable, is determined by the values of the other independent variables, also called regressors. The modelling process basically involves determining parameters of the model, i.e. the weights of regressor variables. The model itself could be linear or non-linear in the parameters. Regression makes a distinction between the response variable and the regressors and is thus generally considered to be a non-symmetric technique.

Here we show how MLR can also be used as a two-class classification tool. The regressor variables are the feature vectors extracted from the

training data. Since we are using regression for classification rather than for modelling, no particular feature is selected as a response variable or expressed in terms of the other features. We posit a separate decision variable, whose value merely indicates the class. The method is thus symmetric in the features. We give below the detailed formulation of MLR as a classification technique.

Suppose there are k features. Let x_{ij} denote the ith observation of feature x_j where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, k$. Let y_i be the ith observed value of the decision variable. Then

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i \tag{1}$$

where the parameters $\beta_j, j = 0, 1, 2, \ldots, k$ are called regression coefficients and ε_i are called error terms or residuals. The regression coefficients are the parameters in the model. Note that the equation is linear in the parameters. The aim is to estimate the values of these parameters from training data. In matrix notation, we have

$$y = X\beta + \varepsilon \tag{2}$$

where y is an $n \times 1$ vector of observations, X is an $n \times p$ matrix of feature values, where p is $k + 1$, β is $p \times 1$ vector of the regression coefficients, and ε is an $n \times 1$ vector of error terms. We may estimate the values of the parameters $\hat{\beta}$ using the least square method. That is, we wish to minimize

$$S(\beta) = \sum_{i=1}^{n} \varepsilon_i^2 = \varepsilon'\varepsilon = (y - X\beta)'(y - X\beta) \tag{3}$$

The least squares estimators must satisfy

$$\frac{\partial S}{\partial \beta}|_{\beta} = -2X'y + 2X'X\hat{\beta} = 0 \tag{4}$$

which leads to

$$\hat{\beta} = (X'X)^{-1}X'y. \tag{5}$$

Observe $(X'X)^{-1}$ exists provided the features are linearly independent.

In order to determine the parameters, we need to know the value of the decision variable on the left hand side of the regression equation. Since the decision variable is not a feature in the system but an additional

variable that merely signifies the class, the value of the decision variable can be chosen arbitrarily subject to the following constraints. In order to ensure adequate separation between the two classes, the values for the two classes must be clearly separated. Also, the choice of the values for the decision variable influences the range of the computed values for the parameters – the value chosen for the decision variable must result in reasonable ranges of values for the parameters, avoiding overflows and underflows in the extreme. Finally, choice of symmetric values for the two classes in the two-class case makes the decision rule and thresholding for rejection simpler. In practice the values are decided after a bit of experimentation with the actual data on hand.

For the two-class classification problem, we use differential features – actual value of each feature is calculated as the difference between the values of the feature for the two classes. Values of the decision variable for the two classes are chosen symmetrically around zero and the parameters are estimated from the training data. A test sample can then be classified as belonging to class C1 or C2 depending upon whether the value of the decision variable is positive or negative. It is possible to reject a sample if the value of the decision variable is too close to zero, that is, closer than a specified threshold. Classification performance can then be specified in terms of precision and recall, or using some combined measure such as the *F*-measure:

$$\text{Re}call = \frac{Ok}{Total} * 100 \tag{6}$$

$$\text{Precision} = \frac{\text{Ok}}{\text{Total-Unknown}} * 100 \tag{7}$$

$$\text{F} = \frac{2\text{PR}}{\text{P} + \text{R}} \tag{8}$$

where *Ok* is the number of test samples that are correctly classified, *Unknown* is the number of test samples that are not classified and *Total* is the total size of the test data. There is usually a trade off between precision and recall and a single combined measure is therefore useful for comparison. *F*-measure is one such measure. The definition showed here gives equal weight for precision and recall.

We have outlined a general method for supervised two-class classification using MLR. The method is conceptually simple and based on sound theoretical foundations. The method is symmetric in the features. As shown in the following sections, techniques also exist for validating the adequacy of the model for a given problem and for evaluating the relative significance of the various features (which can be used for feature selection). Although matrix inversion is required for estimating the values of the parameters, once the model is built classifying objects is very efficient – only computation of the linear regression equation and checking the sign of the decision variable are required. The technique is thus highly suitable for two-class classification problems with a reasonably small number of features.

Adequacy of the Model

There exists a test to determine if there is a linear relationship between the decision variable y and any of the features x_j, $j = 1, 2, \ldots, k$. This test can be viewed as an overall or global test of model adequacy. The null hypothesis (Montgomery et al., 2001) can be defined as:

$$\mathrm{H}_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0. \tag{9}$$

Rejection of null hypothesis implies that at least one of the features x_j, $j = 1, 2, \ldots, k$ contributes significantly to the model.

The analysis of variance (ANOVA) table summarizes information about the sources of variation in the data. Sum of Squares of deviations represents variation present in the data. The sources of variation are due to: Regression Model SS_R and Residuals SS_{Res}. SS_T is the total sum of squares corrected for the mean:

$$SS_T = SS_R + SS_{\mathrm{Res}}. \tag{10}$$

Degrees of freedom, DF, are associated with each sum of squares and are related in the same way. Mean Square is the sum of squares divided by its associated DF (Moore & McCabe, 1993). If the data items are normally distributed, the ratio of the mean square for the regression model to the mean square for residuals follows an F-statistic. This F-statistic tests the null hypothesis that none of the explanatory variables have any effect (that is, the regression coefficients are all zero).

It can be shown that SS_R/σ^2 follows χ^2_k distribution where k is the number of degrees of freedom and equal to the number of regressors in the model. Further, SS_{Res}/σ^2 approximates to χ^2_{n-k-1} and SS_{Res} and SS_R are independent. The F-statistic

$$F_0 = \frac{\frac{SS_R}{k}}{\frac{SS_{\text{Res}}}{n-k-1}} = \frac{MS_R}{MS_{\text{Res}}} \tag{11}$$

follows the $F_{k,n\text{-}k\text{-}1}$ distribution. The observed value of F_0 should be large if at least one $\beta_j \neq 0$. The null hypothesis is rejected if the test statistic F_0 is greater than $F_{k,n\text{-}k\text{-}1}$.

We can also use the p-value obtained from ANOVA to determine whether to reject the null hypothesis. The p-value, also referred to as the probability value or observed significance level, is the probability of obtaining, by chance alone, an F-statistic greater than the computed F-statistic when the null hypothesis is true. The smaller the p-value, the stronger the evidence against the null hypothesis. A p-value of 5% is typically considered low enough to reject the null hypothesis.

Significance of Individual Features

The null hypotheses (Montgomery et al., 2001) for testing the significance of individual regression coefficients, such as β_j, are

$$H_0 : \beta_j = 0. \tag{12}$$

If null hypothesis is not rejected, then this indicates that the regressor x_j can be deleted from the model. The test statistic for this hypothesis is

$$t_0 = \frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 C_{jj}}} \tag{13}$$

where C_{jj} is the diagonal element $(X'X)^{-1}$ of corresponding to $\hat{\beta}_j$. The null hypothesis is rejected if

$$|t_0| > t_{\frac{\alpha}{2},\, n-k-1}.$$

This is a test of the contribution of regressor x_j given the other regressors in the model. This is thus a marginal test – the regression coefficient $\hat{\beta}_j$ depends on all the other regressor variables in the model.

One of the major issues in any classification task is the selection of features. Selecting an optimal subset of features from among the set of potential features is a hard problem. A variety of dimensionality reduction techniques have been explored. The above test for the significance of individual features could be very useful in guiding the feature selection process.

A LANGUAGE IDENTIFICATION SYSTEM FOR INDIAN LANGUAGES

In this section we describe our experiments in language identification among Indian languages using MLR as a classification tool. We start with a discussion on the text representation issues of particular interest to Indian languages.

A text can be considered as a sequence of characters. In alphabetic writing systems such as those used for English and other European languages, a character is simply a letter of the alphabet (or a punctuation mark, a digit or other special symbol), which is typically represented as a single byte in a character encoding scheme such as ASCII. Researchers dealing with such languages have naturally chosen a byte as the basic unit of text. Features such as *n*-grams are defined in terms of bytes. Indian scripts are not alphabetic but rather syllabic in nature. The atomic units of writing are called aksharas – individual bytes have no significance in Indian scripts. A unique feature of Indian languages is that there exists a grammar for the scripts (Murthy, 2005). The script grammar is common and applicable to all Indian languages that use a Brahmi-based script. The script grammar defines the set of valid aksharas. We give below the script grammar for Indian language scripts.

A Grammar for Scripts

Indian scripts are directly based on phonetics – the units of orthography exhibit a more or less one to one correspondence with the spoken sounds. The units of orthography are aksharas, which are essentially C*V syllables where C denotes a consonant and V a vowel. C* segments are also allowed. Since the sound units are largely universal and language independent, ISCII – a common script code standard for Indian language scripts – has chosen to define the script grammar in terms of the basic sound units (Bureau of Indian Standards, 1991). A text encoded in ISCII

encodes the sequences of sound units and is thus a language and script independent representation.

There are a very large number of valid aksharas. The script grammar shown in Figure 1, a finite state machine, defines the set of all valid aksharas. Note that aksharas are defined in terms of more basic units such as vowels, consonants, vowel maatras (vowels that occur in combination with consonants), vowel modifiers (the semi-vowels /M/ and /H/), and halant (which is required to remove implicit vowels in consonants). See Murthy (2005), Negi and Murthy (2004), Muthy and Kumar (2003) for more details. Not all sequences of these basic units are valid. The script grammar defines the valid combinations. The script grammar can also be used to segment texts into aksharas.

Choice of Features

Some researchers have used lists of frequent words to distinguish one language from the other. Comparing with stored lists of frequent words can be very effective for language identification. Our experiments using word lists with Indian languages, not described here, also confirm this point. However, there are several objections to the use of lists of words, affixes etc. As test samples become smaller, chances of finding full words reduce. In small samples, words may be cut and storing lists of

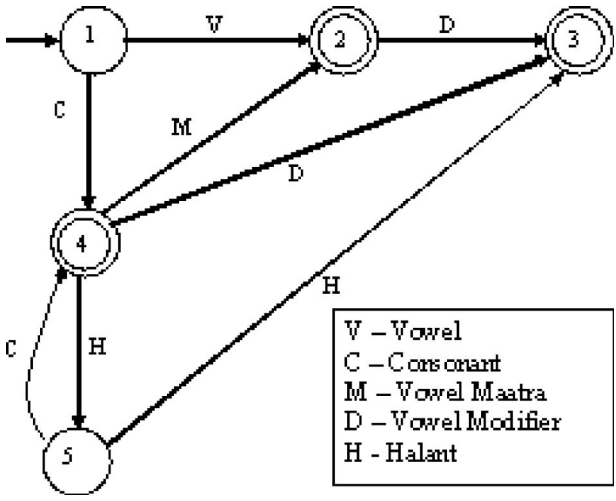

Fig. 1. Script grammar for Indian language scripts.

full words will be of no use. The most frequent words are usually closed class grammatical words such as determiners, prepositions and conjunctions and carry little semantic information. Small text samples exacerbate the bursty nature of texts where such closed class words surround pockets of less common words. It is these less common words that may in fact be more useful for language identification between certain languages than the small function words. Which words to include in a word list is therefore an open question. Lastly, statistical features such as *n*-grams in any case include the information contained in small, frequent words, affixes etc. Given these facts and the desire to build generic, trainable language identification systems, machine learning approaches that depend solely on features extracted from data are preferred. We employ a machine learning approach in our work here. We do not use word lists or any hand crafted rules.

Texts are treated as sequences of aksharas. The script grammar is used to segment texts into aksharas. The features we employ are all expressed in terms of aksharas and sequences of aksharas. Aksharas are smaller units than full words. The number of frequently used aksharas is also smaller than number of words. The number of distinct words is of the order of hundreds of thousands whereas aksharas in common use are in thousands. Our studies have shown that about 5000 aksharas account for more than 99% of all words in all the major Indian languages. Thus the size of the training corpus required will be much smaller if we use aksharas-based features. There is no need to talk in terms of words or morphemes.

After some preliminary exploration, we chose the following features for further exploration:

(a) aksharas which occur frequently in one language but not in the other,
(b) aksharas that occur in word initial position frequently in one language but not in the other,
(c) aksharas that occur in word medial position frequently in one language but not in the other,
(d) aksharas that occur in word final position frequently in one language but not in the other,
(e) akshara bigrams that occur frequently in one language but not in the other,
(f) akshara trigrams that occur frequently in one language but not in the other.

Notice the differential nature of these definitions. We have listed monograms, bigrams and trigrams. We have included positional as well as non-positional features. Obviously, some of these may be partly or wholly subsumed in the others. It may be noted that these features are essentially the same as what linguists call phonotactic constraints (Carson-Berndsen et al., 2004). We shall show below the relative significance of the various features and the effect of choosing only the most promising of these features.

Feature Extraction

One unique feature of our approach is the use of a two level feature extraction process. In the first level, text corpora are used to extract akshara level monograms, bigrams and trigrams. For the current task, the training corpus is simply plain text corpora in the languages concerned. We have used the DoE/CIIL Corpora of Indian languages which include about 3 million word plain text monolingual corpora in each of the major Indian languages (Praksh et al., 2002; Jayaram & Rajyashree, 1996). Only the most frequently occurring and differential features are retained. The result is a set of tables for each pair of languages. The tables simply list the akshara level monograms, bigrams and trigrams that occur frequently in the first language but not in the second, and vice versa. The frequencies themselves are not stored. After this step, full corpora are never used again.

In the second phase, training samples are extracted randomly from the corpora and used for estimating the parameters of the regression model. A training data set consists of random samples containing only a small number of aksharas. Since the features are defined in a differential manner, the actual feature values are obtained by simply counting the occurrences of these features in the training samples. For example, all possible trigrams are extracted from a training sample and each is checked in the feature tables obtained in the first phase. The value of the trigram feature for language L1 is the total number of these trigrams found frequently in L1 but not in L2. Thus the feature values are all integers.

Note that the feature values are not computed directly and solely from the training samples. Instead, they are expressed in terms of the prior knowledge obtained from corpora as encapsulated in the tables obtained in the first phase. Since all we need is plain text corpora and small corpora are sufficient, this two-stage feature extraction is feasible and practicable. The features so extracted can be expected to be more

robust and more reliable than features extracted directly from small training samples.

In order to determine the adequate sizes of training corpora some experiments were conducted. A number of *n*-akshara samples were randomly selected from the total corpus and the regression equation solved to obtain the values of the parameters. Stability of the parameter values was observed for different sample sizes and number of samples used. These preliminary experiments indicated that about 1000 random samples each containing not more than 10 aksharas amounting to a total of only a few thousand words is sufficient for training.

Testing and Results

The parameters of the regression equation were estimated from the training samples randomly extracted from the corpora. Then testing was carried out on test data, also extracted randomly from the rest of the corpus. Each test data set consisted of 1000 test samples. Each sample was analyzed into aksharas and the differential feature values were obtained. The value of the decision variable was computed and the sample classified accordingly. The performance was measured in terms of precision, recall and *F*-measure. This experiment was repeated for different sizes of test data, ranging from five aksharas to 25 aksharas, corresponding roughly to just a couple of words to up to a maximum of about nine words for the languages under consideration. The results are shown in Table 1. Performance has been cross-validated by repeated rounds of training and testing.

It can be seen that an *F*-measure of 98.3% was achieved when test samples were about 10 aksharas in size and the performance went up to 100% when the sample size was increased to about 25 aksharas. These results are comparable to the best results published for other languages of the world.

Table 1. Precision, recall, *F*-measure with respect to mean size of test data (averaged over 1000 samples).

Mean size of test sample	Precision (%)	Recall (%)	*F*-measure (%)
5 aksharas	94.18	93.80	93.99
8 aksharas	97.79	97.40	97.60
10 aksharas	98.40	98.20	98.30
15 aksharas	99.70	99.00	99.35
20 aksharas	99.90	99.60	99.75
25 aksharas	100.00	100.00	100.00

Adequacy of the Model

The overall adequacy of the model for the current task was carried out. Statistical tests for adequacy of the model require that the errors (that is the difference between the computed and assumed values for the decision variable) be normally distributed with mean zero. Figure 2 below, for the Telugu-Hindi pair, shows that the distribution of the error terms is close to normal.

Further, we need to show that there exists a linear relationship between at least some of the regressors. For this, we performed a multiple regression test by considering feature values as independent variables and residuals as dependent variables in order to calculate the *F*-statistic. The result for Telugu-Hindi is shown in Table 2. Since $F_0 >$ the table value of $F_{5,994}$, we can reject the null hypothesis and conclude that there does exist a linear relation between the decision variable and the features, thereby proving the adequacy of the model for the current task.

Test for Significance of Individual Features

Test for the significance of individual features was also carried out. The results are given in Table 3 for the Telugu-Hindi case.

We can observe that the bigram feature is contributing the most. The non-positional monogram feature and word-medial monogram feature

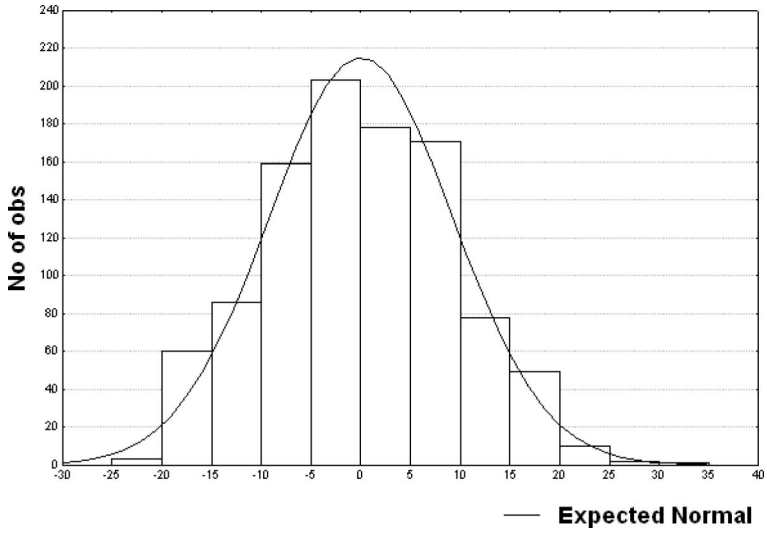

Fig. 2. Distribution of raw residuals.

contribute least. Non-positional monograms are largely subsumed in the positional monograms. Only three of the five features are significant. As we shall see below, trigram feature is useful only when the languages under consideration are very close to one another. For Telugu-Hindi, bigrams are sufficient – trigrams were not used. We repeated the experi-ments with only the three significant features and the results are shown in Table 4.

It may be seen that there is no significant deterioration in the performance of the system. The test for significance of individual features surely helps in the feature selection process.

Table 2. Analysis of variance: Computation of the *F*-statistic for the Telugu-Hindi case.

Source of Variation	Sum of squares	Degrees of freedom	Mean square	F_0	*P*-value	$F_{0.01,5,994}$
Regression	3719.622	5	743.924	16.86607	5.5511e-016	3.32292
Residual	43813.801	994	44.123			
Total	47533.423	999				

Table 3. Significance of individual regression coefficients.

Regressor	t_0	$t_{0.005,2396}$
Monograms	2.576	0.7391
Word initial monograms	2.576	3.1770
Word medial monograms	2.576	1.4328
Word final monograms	2.576	3.5442
Bigrams	2.576	213.1660

Table 4. Performance with respect to mean size of test data – using only three features (averaged over 1000 samples).

Mean size of test sample	Precision	Recall	*F*-measure
5 aksharas	93.99	93.80	93.89
8 aksharas	97.69	97.40	97.55
10 aksharas	98.40	98.10	98.25
15 aksharas	99.70	98.90	99.30
20 aksharas	99.90	99.60	99.75
25 aksharas	100.00	100.00	100.00

Experiments were also performed with different values for the threshold in order to explore the trade-off between precision and recall. Table 5 shows the trade-off between precision and recall. When encountered with the task of identifying the language of a small piece of text, it is possible to initially look for a high-precision, low-recall solution and reduce the threshold value iteratively in case identification fails until a solution is obtained.

The above experiments have been repeated for all pairs of languages among nine major Indian languages for which corpora were available. The results are shown below for test data of 1000 samples extracted randomly from the corpus where each sample includes 10 aksharas. The results are shown separately for within and across the two language families covered – Dravidian and Indo-Aryan (see Tables 6 to 8). It may be observed that the differences between the within-language-family and across-language-family cases are not very drastic. We can, however, see the degree of "closeness" between various language pairs. Thus Hindi and Punjabi are closer than, say, Oriya and Punjabi. It can also be seen that Tamil is quite distinct from all other languages (see Table 9). This is to be expected as Tamil script has a much smaller number of characters compared to any other language considered.

Table 5. Trade-off between precision and recall.

Threshold	Precision	Recall
0	98.30	98.30
1	98.40	98.10
2	98.48	97.20
4	99.56	90.20
6	99.66	87.40

Table 6. Comparison within Dravidian languages (1000 test samples, each 10 aksharas).

Language pair	Precision	Recall	*F*-measure
Telugu-Tamil	100.00	99.93	99.97
Tamil-Malayalam	99.97	99.87	99.92
Telugu-Malayalam	99.80	99.37	99.58
Malayalam-Kannada	99.40	98.77	99.08
Telugu-Kannada	99.32	97.10	98.20
Tamil-Kannada	99.93	99.80	99.87

Table 7. Comparison within Indo-Aryan languages (1000 test samples, each 10 aksharas).

Language pair	Precision	Recall	*F*-measure
Hindi-Bengali	99.48	96.07	97.74
Marathi-Bengali	99.60	98.97	99.28
Oriya-Bengali	99.15	92.97	95.96
Punjabi-Bengali	99.53	98.63	99.08
Punjabi-Oriya	99.77	99.10	99.43
Oriya-Marathi	99.90	98.93	99.42
Punjabi-Marathi	98.96	98.67	98.82
Oriya-Hindi	99.66	96.33	97.96
Punjabi-Hindi	98.99	91.17	94.92
Marathi-Hindi	97.66	97.40	97.53

Table 8. Dravidian vs Indo-Aryan languages (1000 test samples, each 10 aksharas).

Language pair	Precision	Recall	*F*-measure
Kannada-Bengali	99.73	99.30	99.52
Malayalam-Bengali	99.97	99.83	99.90
Tamil-Bengali	100.00	99.97	99.98
Telugu-Bengali	99.90	99.30	99.60
Kannada-Hindi	99.53	98.83	99.18
Malayalam-Hindi	99.80	99.53	99.67
Tamil-Hindi	99.97	99.97	99.97
Telugu-Hindi	99.43	99.00	99.22
Marathi-Kannada	99.43	98.97	99.20
Oriya-Kannada	99.83	99.50	99.67
Punjabi-Kannada	99.77	99.47	99.62
Malayalam-Marathi	99.97	99.77	99.87
Tamil-Marathi	100.00	100.00	100.00
Telugu-Marathi	99.43	99.00	99.22
Oriya-Malayalam	99.93	99.90	99.92
Punjabi-Malayalam	100.00	99.87	99.93
Tamil-Oriya	100.00	100.00	100.00
Telugu-Oriya	99.87	99.53	99.70
Tamil-Punjabi	100.00	99.93	99.97
Telugu-Punjabi	99.77	99.57	99.67

We have also carried out experiments for different sample sizes of test data, ranging from 5 aksharas to 25 aksharas, corresponding roughly to a couple of words to up to about nine words with and without the trigram feature. The results averaged over 1000 test samples are shown for

different language pairs in Table 10 to Table 12. It can be seen that trigram features are required to distinguish between languages of the same family. If the two languages belong to different families, bigrams are sufficient. We thus see quantitative evidence for the validity of these language families.

Table 9. Tamil vs other Indian languages.

Language pair	Precision	Recall	*F*-measure
Tamil-Punjabi	100.00	99.93	99.97
Tamil-Oriya	100.00	100.00	100.00
Tamil-Hindi	99.97	99.97	99.97
Tamil-Bengali	100.00	99.97	99.98
Tamil-Marathi	100.00	100.00	100.00
Tamil-Telugu	100.00	99.93	99.97
Tamil-Malayalam	99.97	99.87	99.92
Tamil-Kannada	99.93	99.80	99.87

Table 10. Performance with and without trigram feature for Hindi-Bengali pair.

	With trigram feature			Without trigram feature		
Test data size (in aksharas)	P	R	F	P	R	F
5	98.64	79.70	88.16	98.64	70.80	82.42
10	99.48	96.07	97.74	99.45	90.30	94.65
15	99.80	99.00	99.40	99.69	96.50	98.07
20	99.90	99.60	99.75	99.70	98.30	98.99
25	100.00	99.80	99.90	99.80	99.40	99.60

Table 11. Performance with and without trigram feature for Telugu-Kannada pair.

	With trigram feature			Without trigram feature		
Test data size (in aksharas)	P	R	F	P	R	F
5	98.22	88.30	93.00	97.91	79.70	87.87
10	99.32	97.10	98.20	98.94	92.90	95.82
15	99.70	99.10	99.40	98.77	96.50	97.62
20	99.80	99.70	99.75	99.70	98.30	98.99
25	100.00	99.70	99.85	99.90	99.70	99.80

Table 12. Performance with and without trigram feature for Kannada-Bengali pair.

	With Trigram Feature			Without Trigram Feature		
Test data size (in aksharas)	P	R	F	P	R	F
5	99.05	93.60	96.25	95.89	95.70	95.80
10	99.73	99.30	99.52	99.00	99.00	99.00
15	100.00	99.90	99.95	99.60	99.30	99.45
20	100.00	100.00	100.00	99.90	99.80	99.85
25	100.00	100.00	100.00	100.00	99.90	99.95

Aksharas are Fundamental Units of Orthography

Given that there is a grammar at the level of scripts and that there are valid and invalid sequences making up aksharas, aksharas would be the natural choice as basic units of writing. The notion of a character has to be clearly understood in the context of Indian scripts. It is not appropriate to view vowels and consonants along with other punctuation marks and special symbols and call them characters. Sequences of such symbols, typically coded as bytes inside computers, could be ungrammatical. These invalid sequences can never occur in any language, not even in proper nouns or acronyms. Ungrammaticality is the not the same as spelling error. Vowels, consonants or bytes in general are not appropriate units in Indian languages. Aksharas are the fundamental units.

In order to obtain empirical support for this argument, we have conducted experiments taking bytes as units. Under otherwise identical set up, we find that using bytes as the basis consistently leads to significant deterioration in performance as can be seen from Table 13 below. The results shown are based on repeated experiments with random samples from a corpus. This provides experimental evidence to support our view that aksharas should be taken as basic units of orthography in Indian scripts, not characters or bytes.

CONCLUSION

In this paper language identification has been formulated as a generic machine learning problem, a supervised classification task in which features extracted from a training corpus are used for classification. We

Table 13. Akshara-level features vs byte-level features (bigrams and trigrams) (averaged over 1000 samples).

	Akshara-based			Byte-based		
Language pair	P	R	F	P	R	F
Hindi-Bengali	99.48	96.07	97.74	92.80	85.03	88.74
Kannada-Bengali	99.73	99.30	99.52	99.09	98.36	98.72
Kannada-Hindi	99.53	98.83	99.18	96.31	92.09	94.15
Malayalam-Bengali	99.97	99.83	99.90	99.16	98.07	98.61
Malayalam-Hindi	99.80	99.53	99.67	98.12	95.67	96.88
Malayalam-Kannada	99.40	98.77	99.08	96.54	91.69	94.05
Malayalam-Marathi	99.97	99.77	99.87	99.17	97.31	98.23
Marathi-Bengali	99.60	98.97	99.28	94.21	92.03	93.10
Marathi-Hindi	97.66	97.40	97.53	82.69	82.22	82.46
Marathi-Kannada	99.43	98.97	99.20	96.74	96.55	96.65
Oriya-Bengali	99.15	92.97	95.96	90.20	77.07	83.12
Oriya-Hindi	99.66	96.33	97.96	96.71	92.03	94.31
Oriya-Kannada	99.83	99.50	99.67	98.64	97.52	98.08
Oriya-Malayalam	99.93	99.90	99.92	98.96	97.93	98.44
Oriya-Marathi	99.90	98.93	99.42	92.97	92.97	92.97
Punjabi-Bengali	99.53	98.63	99.08	98.12	95.77	96.93
Punjabi-Hindi	98.99	91.17	94.92	93.35	93.17	93.26
Punjabi-Kannada	99.77	99.47	99.62	99.63	98.79	99.21
Punjabi-Malayalam	100.00	99.87	99.93	100.00	99.83	99.92
Punjabi-Marathi	98.96	98.67	98.82	94.16	92.23	93.18
Punjabi-Oriya	99.77	99.10	99.43	99.10	98.33	98.71
Tamil-Bengali	100.00	99.97	99.98	100.00	99.50	99.75
Tamil-Hindi	99.97	99.97	99.97	99.87	99.87	99.87
Tamil-Kannada	99.93	99.80	99.87	99.60	99.30	99.45
Tamil-Malayalam	99.97	99.87	99.92	98.63	98.53	98.58
Tamil-Marathi	100.00	100.00	100.00	100.00	96.47	98.20
Tamil-Oriya	100.00	100.00	100.00	100.00	99.73	99.87
Tamil-Punjabi	100.00	99.93	99.97	100.00	100.00	100.00
Telugu-Bengali	99.90	99.30	99.60	97.80	97.60	97.70
Telugu-Hindi	99.43	99.00	99.22	89.82	89.73	89.78
Telugu-Kannada	99.32	97.10	98.20	90.66	79.17	84.53
Telugu-Malayalam	99.80	99.37	99.58	96.63	96.63	96.63
Telugu-Marathi	99.43	99.00	99.22	94.33	87.64	90.86
Telugu-Oriya	99.87	99.53	99.70	97.47	97.47	97.47
Telugu-Punjabi	99.77	99.57	99.67	98.77	96.50	97.62
Telugu-Tamil	100.00	99.93	99.97	99.63	99.63	99.63

have formulated the two-class supervised learning problem using MLR and applied this to the problem of language identification among Indian languages. Techniques for verifying the adequacy of the model and for

verifying the contribution of individual features have been included. This formulation has been applied for pair-wise language identification among nine major languages. The results obtained are comparable to the best published results for other languages of the world. Although there are a large number of languages in India and Indian language documents are very often multi-lingual, Language identification from small text samples had remained a largely unexplored problem in the Indian context.

We have shown that the method works both across and within language families, although a more sophisticated feature set is required when the languages under consideration are very similar. This idea can be extended further to study other kinds of variations among languages or language families as also to uncover universal, language-invariant features in a quantitative way.

We have also argued for aksharas as the fundamental units of writing in Indian scripts, not characters or bytes. Our experimental results support this view.

ACKNOWLEDGEMENT

The research work reported in this paper was supported in part by the University Grants Commission under the UPE Scheme.

REFERENCES

Adams, G., & Resnik, Ph. (1997). A language identification application built on the Java client-server platform. In J. Burstein & C. Leacock (Eds), *From Research to Commercial Applications: Making {NLP} Work in Practice* (pp. 43–47). Somerset, New Jersey: Association for Computational Linguistics.

Allison, P. D. (1999). *Multiple Regression: A Primer*. Thousand Oaks, CA: Pine Forge Press.

Beesley, K. (1988). Language Identifier: A computer program for automatic natural-language identification of on-line text. *Proceedings of the 29th Annual Conference of the American Translators Association*, 47–54.

Bureau of Indian Standards (1991). *Indian Script Code for Information Interchange – ISCI, – IS 13194: 1991*. New Delhi, India.

Carson-Berndsen, J., Kelly, R., & Neugebauer, M. (2004). Automatic acquisition of feature-based phonotactic resources. *Proceedings of the Workshop of ACL Special Interest Group on Computational Phonology – SIGPHON*, Barcelona, Spain, 27–34.

Combrinck, H., & Botha, E. (1995). Automatic language identification: Resisting complexity. *South African Computer Journal*, *27*, 18–26.

Constable, P., & Simons, G. (2000). Language identification and IT: Addressing problems of linguistic diversity on a global scale. *SIL Electronic Working Papers 2000–2001*.

Dunning, T. (1994). Statistical identification of language. *Technical Report, CRL MCCS-94-273*, New Mexico State University.

Giguet, E. (1995a). Multilingual sentence categorization according to language. *Proceedings of the European Chapter of the Association for Computational Linguistics SIGDAT Workshop from Text to Tags: Issues in Multilingual Language Analysis*, Dublin, Ireland, 73–76.

Giguet, E. (1995b). Categorization according to language: A step toward combining linguistic knowledge and statistical learning. In *4th International Workshop of Parsing Technologies*, Prague, Karlovy Vary, Czech Republic, September, 20–24.

Glantz, S. A., & Slinker, B. K. (2000). *Primer of Applied Regression and Analysis of Variance* (2nd edition). New York: McGraw-Hill.

Jayaram, B. D., & Rajyashree, K. S. (1996). Corpora in Indian languages. In *Workshop on Indian Language Corpora*, CIIL, Mysore.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2001). *Introduction to Linear Regression Analysis*. New York: John Wiley & Sons, Inc.

Moore, D., & McCabe, G. (1993). *Introduction to the Practice of Statistics*. New York: W. H. Freeman and Company.

Murthy, K. N. (2005). *Natural Language Processing – An Information Access Perspective*. Bangalore: Sarada Ranganathan Endowment for Library Science.

Murthy, K. N., & Kumar, G. B. (2003). Language Identification from Small Text Samples. *Technical Report, LERC/UoH/2003/1*, Department of Computer and Information Sciences, University of Hyderabad, India.

Muthusamy, Y. K., Barnard, E., & Cole, R. A. (1994). Automatic language identification: A review/tutorial. *IEEE Signal Processing Magazine*, *11*(4), 33–41.

Negi, A., & Murthy, K. N. (2004). Issues of document engineering in the Indian context. *Technical Report, LERC/UoH/2004/1*. Department of Computer and Information Sciences, University of Hyderabad, India.

Piotrowski, M. (1997). Statistical language identification for NLP-supported full text retrieval. *Technical Report, CLUE-TR-3*.

Prager, J. M. (1999). Linguini: Language identification for multilingual documents. *Journal of Management Information Systems*, *16*(3), 71–101.

Rao, K. P., Bharati, A., Sangal, R., & Bendre, S. M. (2002). Basic statistical analysis of corpus and cross comparison among corpora. In *Proceedings of the Recent Advances in Natural Language Processing (ICON-2002)*, 121–129.

Sinha, R. M. K. (1984). Computer processing of Indian languages and scripts – potentialities and problems. *Journal of the Institute of Electronic and Telecommunications Engineers*, *30*(6), 133–149.

Wechsler, M., Sheridan, P., & Schauble, P. (1997). Multi-language text indexing for Internet retrieval. In *Proceedings of the 5th RIAO Conference, Computer-Assisted Information Searching on the Internet*, Montreal, Canada, 217–232.

Xafopoulos, A., Kotropoulos, C., Almpanidis, G., & Pitas, I. (2004). Language identification in web documents using discrete hidden Markov models. *Pattern Recognition*, *37*(3), 583–594.