Addressing challenges in automatic Language Identification of Romanized Text

Kosuru Pavan
Language Technologies Research Centre,
IIIT Hyderabad, India
pavank@research.iiit.ac.in

Niket Tandon
DataBases and Information Systems
Max-Planck-Institut fr Informatik
ntandon@mpi-inf.mpg.de

Vasudeva Varma
Language Technologies Research Centre,
IIIT Hyderabad, India
vv@iiit.ac.in

Abstract

Due to the diversity of documents on web, language identification is a vital task for web search engines during crawling and indexing of web documents. Among the current challenges in language-identification, the unsettled problem remains identifying Romanized text language. The challenge in Romanized text is the variations in word spellings and sounds in different dialects. We propose a Romanized text language identification system (RoLI) that addresses these challenges. RoLI uses an n-gram based approach and also exploits sound based similarity of words. RoLI does not rely on language intensive resources and is robust to Multilingual text. We focus on five Indian languages: Hindi, Telugu, Tamil, Kannada and Malayalam. Over the five languages, we achieve an average accuracy of 98.3%, despite the spelling variations as well as sound variations in Indian languages.

1 Introduction

For web search engines, Language identification is an important task during indexing. However, Web pages barely provide reliable meta-data that indicates the language of the web page. Several web pages are multilingual in nature and even worse, many web documents contain text portions in Romanized text. For users, it is rarely a problem to identify the language of a document as long as they understand the language. However, language is a barrier for user access when they encounter a page in an unknown language. They would want to translate the page automatically with an online tool. In this scenario, the source language needs to be identified.

Language Identification is labeled as solved problem from long time but challenges remain with the growing Internet documents. Different classification models that use the n-gram features have been proposed. (Cavnar and Trenkle, 1994) used an out-of-place rank order statistic to measure the distance of a given text to the n-gram profile of each language. (Dunning, 1994) proposed a system that uses Markov Chains of byte n-grams with Bayesian Decision Rules to minimize the probability error. (Grefenstette, 1995) simply used trigram counts that are transformed into probabilities, and found this superior to the short words technique. (Sibun and Reynar, 1996) used Relative Entropy by first generating n-gram probability distributions for both training and test data, and then measuring the distance between the two probability distributions by using the Kullback-Liebler Distance. (Poutsma, 2001), developed a system based on Monte Carlo Sampling. (Radim and Kolkus, 2009) constructs language models based on word relevance to address the limitations in n-gram method. Linguini, a system proposed by (Prager, 1999), combines the word-based and n-gram models using a vector-space based model and examines the effectiveness of the combined model and the individual features on varying text size. Similarly, (Lena Grothe and Nrnberger, 2008) combines both models using the ad-hoc method of (Cavnar and Trenkle, 1994), and also presents a comparison study. Recently, in (Hammarstrm, 2007), which proposes a model that uses a frequency dictionary together with affix information in order to identify the language of texts as short as one word. (Hughes et al., 2006) surveyed the previous work in this area and suggested that the problem of language identification for written resources, although well studied, has many open challenges that require a more systematic and collaborative study. (Pingali et al., 2006) proposed a system for searching multi-

Proceedings of ICON-2010: 8th International Conference on Natural Language Processing, Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2010

script and multi-encoded Indian language content on the web.

Another largely un-addressed problem in Language identification is Romanized Text Language Identification. In linguistics, romanization or latinization, is the representation of a written word or spoken speech with the Roman (Latin/English) alphabet, or a system for doing so, where the original word or language uses a different writing system. As stated in (Naveen Gaba and P.K.Saxena, 2003), Romanized language uses literal units that are a string of different lengths created on the basis of their phonetic sounds. Each language has its own sets of vowels and consonants. A consonant followed by a vowel is called literal unit. Each consonant independently also forms a literal unit. Romanized language uses these literal units to form words. Methods of romanization include transliteration, for representing written text, and transcription, for representing the spoken word. Here we consider only transliteration text identification problem. This is very hard problem because of the variations in word spellings and sounds in respective dialects in that particular language.

Language identification of Romanized text is a little explored domain. Work by (Naveen Gaba and P.K.Saxena, 2003) comes closest to this effort. They use language specific features to discriminate in the Romanized form for Bengali, Hindi and Punjabi. The features used are, Index of coincidence, Frequency of letters, vowel or consonants percentage, Vowels contacting consonants and vice versa, Frequency of vowel-vowel-vowel triplet, frequency of four consonants occurring together, frequency of deviation, matching index, Frequency of letters with highest contact on both sides are the feature set. The results obtained by this method were not good for a combination of Punjabi and Hindi. Another limitation is the scope of character sets for which the system worked. For instance, (Martins and Silva, 2005) accepts that the system will not work for higher-grade encryption schemes.

We propose RoLI:Romanized text Language Identification system, that can identify the language of Romanized text. We address the problems of encoding schemes, with a generic system and with a better performance. RoLI also identifies language of Romanized text containing a few English words, multi language documents. We select five Indian languages along with English as a multi-lingual text in several documents. Out of the five languages chosen, Hindi one of the most widely spoken Indian languages across the nation, followed by Telugu, Tamil, Kannada and Malayalam. Due to the diverse dialects in Indian languages it is very difficult to recognize them using dictionary methods. Hindi itself has more than 10 dialects and the romanization of the languages changes with the dialects. Almost all the Indian languages are inflectional language (except Hindi) (Paik and Parui, 2008). This makes it difficult to apply any statistical stemmers or morphers to apply the dictionary method for language identification.

In this paper, we discuss different methods to romanize the text. The major contributions of our work are:

- First attempt to identify Romanized text language for several Indian languages
- Less dependence on language intensive resources
- Handling spelling and phoneme variations
- Handling multiple languages text
- Developing corpus suitable for Romanized Text Language Identification.

2 RoLI- A system for Language Identification of Romanized text

Romanization is extensively used by native users on the Internet to express in their mother tongue. e.g. In Figure 1 the author uses Romanization (Telugu). There have been many standard Romanization techniques for many languages in the web. For example `rõmaji` is a Romanized system for Japanese, `Hanyun Pinyun` is Romanized system for Chinese, `Revised Romanization of Korean` is the official romanization system in South Korea. Indian languages do not have any standard technique for such representation. Therefore in the Romanization of Indian Language text, different people use different spellings for same words. This is also due to variation in the dialects of those language users. For example the word *ela vunnaru*(Telugu) can be written using different spelling variations, *elaa vunnaru* or *elaa vunaroo* or *ela vunaru* etc. Therefore we need a system which can identify both the phonetic as well as

spelling differences. In our system we are addressing both the variations using combination of Classic Soundex algorithm (Rosen, 1994), and Levenshtein distance (Navarro, 2001) for language identification of Romanized text.

We observe the following complexities in Romanized text:

- **Spelling or word variations**: Based on their comfort and usage different people tend to use different spellings for same word. This happens because of dialects and different people tend to pronouns same word differently. So they write it as they pronounce it.
- **Different phonemes sounds:** There are no standard romanization rules for each language. People tend to use their own based on the phone sounds. For example, the word *telusa* (Telugu) can be written by different people differently *telusoo*, *telsu*, *telsoo*, *telisu* are different phoneme sounds of the same word based on different dialects. These variations can be in a large number, depending upon the word.
- **Closely related words:** In Indian languages, several words are derived from single root word. For example, the word *aap* (Hindi) can have different forms like *aapko*, *aapke*; all these words are derived from single root word. If we are able to get the language of root word its easy to identify all the closely related words. It is not feasible to collect all the words in any Indian language to form a dictionary and search in the dictionary.
- **Mix of English and Romanized text:** From the figure 1 it can be observed that the blog users use English text in their blogs irrespective of language of remaining text, because English is the most widely used language across globe. People speaking different language use English words intermittently. Therefore, it is common to include English text in Romanized text of a language. Also for web pages, English is widely used language, so they tend to have English plain text as the title and remaining text is in some other language in its Romanized form.

RoLI addresses these complexities of Romanized text in order to perform language identification of Romanized text.

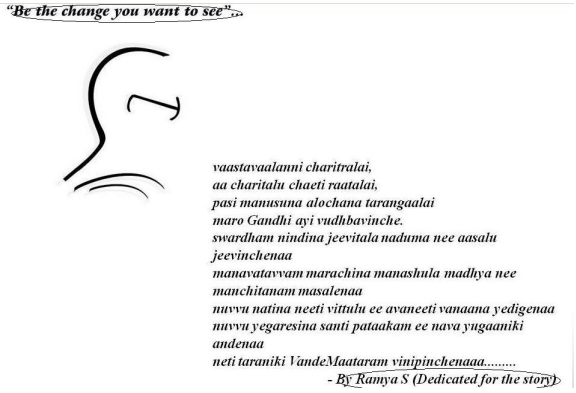

Figure 1: Blog with both Romanized(Telugu) and English text

3 Approach

We generate the profile for the input document as in language model. The method is as follows, for a given Document (D_i) we tokenize the document into words $w_1, w_2, w_3, ..., w_n$. We compute the similarity of document profile D_i from the language profile L_i Using the following formula we compute the similarity measure

$$Similarity(D_i, L_i) \propto \left(\frac{1}{\sum_{k=1}^{n} LD(w_k, L_i)} \right) \quad (1)$$

here λ is proportionality constant, LD is levenshtein distance and L_i, ith profile in n language profiles.

RoLI identifies the language of the document by computing the similarity distance of the input document from the model generated for each Language. RoLI make use of the similarity measures to calculate the relatedness between the input profile and language profile. The rank-order statistics is used for calculating the similarity measures. After tokenizing and computing language profile for input document, the relatedness between its entries to the language profile entries is calculated by using the Formula 1. This results in number of entries belonging to the each language and by computing the percentage of entries related to the language, we get the language of the input document.

Next, we explain the language model for Identifying the Romanized text.

3.1 Language Model for Language Identification

RoLI is trained in order to identify the language of the input text. We form a model with Romanized

word, Phoneme sound and Language of that word as features. The method generates language specific profiles which contains most frequent words in the corpus, and their phonetic sound. We used Soundex algorithm to generate the phonetic sound. It only encodes consonants, the primary goal is to capture closely pronounced words, homophones to be encoded to the unique representation in order to match despite minor differences in spelling. This helps in finding the same representation for all forms of same word. For example the words *akkada*, *akkadiki*, *akkadaku*, *akkadika* in Telugu belong to the same root word *akkada*. Soundex gives a unique representation for all these words(A230). We also used Levenshtein distance to compute the near match for a word from homophones and identify the closest form of that word from the existing entries. The following shows the entries in the language model for Romanized text

$$< word_I, Phonetic(word_I), lang_{(word_I)} >$$

here $word_I$, word to be stored in the model.

The system is trained with 80% of collected documents in DataSet-2 for each language. We build the model for all the languages with labeled data as training data.

4 Evaluation

The following section explains the experimental process of our data collection, later we experiment over this data collected using different evaluation methods.

4.1 Data Collection

Romanized text is very difficult to garner from the web as Search Engines identify the Romanized text as English text. To the best of our knowledge, no search engine ever tried to identify language of Romanized text. We developed a system to extract Romanized data from the web. We collected two different kinds of data for our model, both for training as well as testing. This contains (1) Blogs in many regional language use the Romanized notation in their local language, (2) Google-Script-Converter(GSC), online script-converter for many Indian languages, for preparing rule-based data for our system.

Yahoo and Google Search Engines were used to collect the Blogs and forum data. We prepared relevance feedback model to retrieve Romanized blog data and web pages. For each language, the common terms are extracted, and passed as queries to search engines. As stated by (Ruthven and Lalmas, 2003), we add some additional terms to the query based on their probability of occurrence in relevant document model in the whole collection. The query is expanded with the most frequent terms from top 7-8 documents, by adding them to basic query. This query is passed again to the search engines and the top n-documents in each language are collected. Next, identify the top ranked documents for the modified query and collect that data. This is a semi-automated process, ranking of top n documents for relevance feedback involved slight user intervention for better data processing.

Table 1: Documents collected in two datasets

Language	DataSet1	DataSet2
Telugu	500	9500
Tamil	500	9500
Hindi	500	9600
Kannada	250	8500
Malayalam	250	9300

Table-1 shows the number of documents using this process, labeled as DataSet1. Small Numbers in Table-1 shows the sparsity of this kind of data over Internet and difficulty of finding them using current search engines.

GSC converts all the unicode content in Indian languages to the Romanized format based on their pronunciations. It uses rules for each character, to convert them to Romanized text. This is also called transliteration of the Indian Language text. For each character in the unicode text GSC apply character to character mapping, it generates machine prepared Romanized text for the training module of RoLI. GSC takes the Indian language web page as input and return the Romanized format of the text present in it. It uses unique character to character mappings rules to convert the text into Romanized format, we consider this as artificial data. We collected more than 9000 URLs for each language and generated Romanized data from them using GSC. We collected the text from all the extracted text documents. Table-1, shows the number of documents we collected using this process labeled under DataSet2. We follow the 80%-20% rule for training and testing process.

It is difficult and time consuming to evaluate language identification systems using the web

Table 2: Monolingual Evaluation of RoLI

Lang	2-4 length documents	Average documents
Telugu	98.1	99.1
Tamil	97.7	98.7
Hindi	98.1	98.8
Malayalam	97.1	98.0
Kannada	97.4	97.9

Table 3: Multilingual evaluation of RoLI for 1000 documents of DataSet2 and DataSet1

LangPair	DataSet1	DataSet2
Te-en	98.3	99
Ta-en	98.1	98
Hi-en	98.7	99
Ka-en	95.1	95
Ml-en	94.1	96

data. In order to cover all the variety of data available online we followed the above three different evaluation measures for evaluating the performance of RoLI.

4.2 Monolingual Romanized text

We use 20% of the collected web documents in Table 1 as the test data in this monolingual evaluation of Romanized text. The test data here is completely different from the data we used for language model preparation earlier. We collected 2500 documents for each language and tested on the system. To our surprise, RoLI achieves 100% accuracy for documents of more than 15 words size. Most Language identifiers work well for the documents of large size, they come with the assumption that document or input should be more than 4 words length. We show that our model even works well for documents of small length (2-4 words).

There are several phonetically similar words in Southern Indian Languages. Table-2 shows that Southern Indian languages like Telugu, Tamil, Kannada and Malayalam perform very closely as the length of the document is small, this property is also the cause of some commonly spelled words. It can be observed that the precision increase Table-2 as the length of the documents increases (>4words).

4.3 Multilingual Romanized text

We select language pairs from all the five languages considered earlier. Here we consider bilingual data as multilingual, because we found that it is very unlikely that Romanized text contains more than two language text. We found that English is used along with Romanized text regularly. We also excluded the case where speaker of one language attempts to write in some other language without knowing its script. Table 4 shows performance of our method for multilingual data where Te-en, Ta-en etc. represents the language pair and Te and Ta are dominating languages in those documents respectively.

We applied manual evaluation for multilingual Romanized text identification for blog data i.e. DataSet1. RoLI's precision drops by only 1% in identification of Romanized text language, as shown in Table 3.

Table 4: Performance of RoLI over baseline

Lang	DataSet1		DataSet2	
	n-gram	RoLI	n-gram	RoLI
Telugu	83	97.8	94.0	99.1
Tamil	85	98	94.1	98.7
Hindi	90	96	95.5	98.8
Kannada	82	93	93.4	97.9
Malayalam	79.3	94	92	98.0

4.4 Performance over baseline

Our third evaluation measure is against the baseline system. We used state-of-the-art language identification method, n-gram method, as baseline for our evaluation. In order to use n-gram method one needs language profiles for each language. Language profiles were generated for each language from the training data in DataSet2. We evaluated the system on the Romanized data collected from web and the Romanized data generated using GSC. The precision was calculated using the n-gram method and using RoLI. Table-4 shows the clear improvement of our method over the n-gram method. From the Table -4 we can also observe that n-gram method shows better results for Hindi than remaining languages for both the datasets. Pronunciation of south Indian languages are closely related; Telugu, Kannada and Malayalam, Tamil have several phonetically similar words. This affects the performance of n-gram method. Soundex and Levenshtein distance were used to distinguish this closeness in the languages.

5 Conclusions

In this study, we proposed RoLI, a system to identify the Romanized text for different Indian languages. To the best of our knowledge RoLI is the first system that addresses language identification of Romanized text in detail for several Indian Languages. Indian Languages have varying dialects in different regions; we handle diversity of people's dialect by addressing word and phoneme variations. Our method can be applied to any language using less language specific resources. RoLI achieved a high accuracy of 98.3% in experiments conducted over five Indian language web pages containing a mix of these languages. RoLI is also very useful for short language text compared to other applications. Our work finds its application in the language identification of documents during indexing, this will enable search for Romanized documents as well, especially blogs. In future, we would like to experiment with machine learning approach to develop a classifier taking different language properties as features. We also plan to identify the originated language of the Named Entities with some machine learning approaches.

References

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. pages 161–175, Las Vegas,USA.

Ted Dunning. 1994. Statistical identification of language. Technical report.

G. Grefenstette. 1995. Comparing two language identification schemes. In *In Proceedings of Analisi Statistica dei Dati Testuali (JADT)*.

Harald Hammarstrm. 2007. A fine-grained model for language identification. In *In Proceedings of iNEWS-07 Workshop at SIGIR 2007*, pages 14–20, Amsterdam, july. ACM.

Baden Hughes, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew Mackinlay. 2006. Reconsidering language identification for written language resources. In *Proceedings of LREC2006*, pages 485–488.

Ernesto William De Luca Lena Grothe and Andreas Nrnberger. 2008. A comparative study on language identification methods. In *Proceedings of LREC 2008*, Marrakech, Morocco, May.

Bruno Martins and Mário J. Silva. 2005. Language identification in web pages. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 764–768, Santa Fe, New Mexico. ACM.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88.

Sarvajeet kour Naveen Gaba and P.K.Saxena. 2003. In *ICAPR '03: Identification of encryption schemes for Romanized Indian Languages in proceedings of ICAPR, 2003*, pages 164–168.

Jiaul H. Paik and Swapan K. Parui. 2008. A simple stemmer for inflectional languages. Technical report.

Prasad Pingali, Jagadeesh Jagarlamudi, and Vasudeva Varma. 2006. Webkhoj: Indian language ir from multiple character encodings. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 801–809, New York, NY, USA. ACM.

Arjen Poutsma. 2001. Applying monte carlo techniques to language identification. In *In Proceedings of Computational Linguistics in the Netherlands (CLIN*, pages 179–189. Rodopi.

John M. Prager. 1999. Linguini: Language identification for multilingual documents. In *Journal of Management Information Systems*, pages 1–11.

Radim and Milan Kolkus. 2009. Language identification on the web: Extending the dictionary method. In *CICLing '09*, pages 357–368, Berlin, Heidelberg. Springer-Verlag.

Jeff Rosen. 1994. A simple soundex program. *C/C++ Users J.*, 12(9):49–51.

Ian Ruthven and Mounia Lalmas. 2003. A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.*, 18(2):95–145.

Penelope Sibun and Jeffrey C. Reynar. 1996. Language identification: Examining the issues.