Automatic Language Identification and Content Separation from Indian Multilingual Documents Using Unicode Transformation Format

Rajnish M. Rakholia and Jatinderkumar R. Saini

Abstract In Natural Language Processing (NLP), language identification is the problem of determining which natural language(s) are used in written script. This paper presents a methodology for Language Identification from multilingual document written in Indian language(s). The main objective of this research is to automatically, quickly, and accurately recognize the language from the multilingual document written in Indian language(s) and then separate the content according to types of language, using Unicode Transformation Format (UTF). The proposed methodology is applicable for preprocessing step in document classification and a number of applications such as POS-Tagging, Information Retrieval, Search Engine Optimization, and Machine Translation for Indian languages. Sixteen different Indian languages have been used for empirical purpose. The corpus texts were collected randomly from web and 822 documents were prepared, comprising of 300 Portable Document Format (PDF) files and 522 text files. Each of 822 documents contained more than 800 words written in different and multiple Indian languages at the sentence level. The proposed methodology has been implemented using UTF-8 through free and open-source programming language Java Server Pages (JSP). The obtained results with an execution of 522 Text file documents yielded an accuracy of 99.98 %, whereas 300 PDF documents yielded an accuracy of 99.28 %. The accuracy of text files is more than PDF files by 0.70 %, due to corrupted texts appearing in PDF files.

Keywords Gujarati • Indian language • Language Identification (LI) • Natural Language Processing (NLP) • Unicode Transformation Format (UTF)

R.M. Rakholia (✉)
R K University, Rajkot, Gujarat, India
e-mail: rajnish.rakholia@gmail.com

J.R. Saini
Narmada College of Computer Application, Bharuch, Gujarat, India
e-mail: saini_expert@yahoo.com

S.C. Satapathy et al. (eds.), *Proceedings of the International Conference on Data Engineering and Communication Technology*, Advances in Intelligent Systems and Computing 468, DOI 10.1007/978-981-10-1675-2_37

1 Introduction

Language detection and language identification plays an important role in the field of Natural Language Processing (NLP). On the internet, written text is available in number of languages other than English and many document and web pages contain mix language text (more than one language in same document or same web page) such as Gujarati, Hindi, and English [1, 2].

1.1 Language Identification

Language identification is the task of automatically detecting the language present in a document based on the written text of the document and character encoding used in web page. Detecting multilingual documents and texts is also important for retrieve linguistic data from the internet. Language identification is the problem of classifying words and characters based on its language, it can be used for preprocessing stage in many applications (viz. parsing raw data, tokenizing text) to improve the quality of input data based on language specific model.

1.2 State of the Art (Language Identification)

Many methods and techniques with very high precision are available to identify popular languages in the world like English, German, Chinese, etc., from multilingual documents, but it cannot be applicable directly on resource poor languages due to its morphological variance and complex structure of framework such as Gujarati, Punjabi, and other Indian language.

1.3 Unicode Transformation Format

Unicode Transformation Format (UTF) is a standard character set which is used to display the character in proper format, which is written using different languages like: Gujarati, Hindi, Tamil, etc. These all are Indian languages which is not possible to display each character using American Standard Code for Information Interchange (ASCII), but it is possible to use in English. There are three different Unicode representations: 8-bit, 16-bit, and 32-bit encodings. UTF is supporting more diverse set of characters and symbols for different languages. We have used UTF for Indian language only, and it is mostly use in web technology and mobile application [3].

1.4 Essential of Language Identification

Many number of multilingual documents available on the internet in digital form in multilingual country like India. Different language has different framework and grammatical structure. Therefore, its need to automation tools which can identify the language(s) from written document and apply appropriate tool for further processing based on language(s) detect in document. A number of applications such as POS-Tagging, information retrieval, search engine, machine translation, accessibility of webpage, and other language processing activities require Language Identification as preprocessing step in multilingual environment.

1.5 Tools for Language Identification

Table 1 lists, number of tools (freely and commercially) available for automatic language identification.

2 Related Work on Language Identification

According to Verma and Khanna (2013) audio speech contains various information like gender, language spoken, emotion recognition, and phonetic information. They presented automatic language identification system using k-means clustering on MFCCs for features extraction and Support Vector Machine for classification. They

Table 1 Language identification tools

Sr. No.	List of tools available	Number of supported languages
1	Languid	72
2	Textcat	69
3	C# package for language identification of Microsoft	52
4	Xerox MLTT language identifier	47
5	Rosette language identifier by Basis Technology	30
6	SILC/Alis	28
7	Lid	23
8	Collexion	15
9	Stochastic language identifier	13
10	Langwitch by morphologic	7
11	Lextek language identifier	Many
12	Language identification program by ted dunning	2

tested proposed system on custom speech database for three Indian languages English, Hindi, and Tibetian. They achieved average classification accuracy of 81 % using small duration speech signals [4].

Anto et al. (2014) developed speech language identification system for five Indian languages, English (Indian), Hindi, Malayalam, Tamil, and Kannada. They had not used publicly available speech databases for these languages, but they created manually dataset by downloading YouTube audio file and remove the non-speech signals manually. They tested this system using created dataset consisting of 40 utterances with duration of 30, 10, and 3 s, in each of five target languages. They used 3, 4, and n g language models to implement this system. After experiment of this system, result shown that the use of 4 g language models can help enhance the performance of LID systems for Indian languages [5].

Yadav and Kaur S (2013) presented work related to identify different 11 regional Indian languages along with English from OCR corrupted text. They used distance measure-based metric to correct the text, naive Bayesian classification to identify the language of corrupted text and different n-gram model to represent the language. They tested this technique on different length of text, different n-gram (3, 4, and 5 g) language models and different percentage of corrupted texts [6].

Padma M et al. (2009) used profile features for language identify from multilingual document written in Indian languages. They have proposed to work on the prioritized requirements of a particular region, for instance in Karnataka state of Indian, English language used for general purpose, Hindi language for National importance, and Kannada language for State/Regional importance. They proposed very common concept in which they used bottom and top profile of character to identify languages from Indian multilingual document. In experimental setup they used 600 text lines for testing and 800 text lines for learning. They achieved average 95.4 % of accuracy [7].

Chanda S et al. (2009) proposed a scheme, to identify Thai and roman languages written in single document. They used SVMs-based method in proposed system to identify printed character at word level. They obtained accuracy of 99.62 %, based on the experiment of 10000 words [8].

According to Saha S et al. (2012), they studied and compared various feature reduction approaches for Hindi and Bengali Indian languages. They also studied different dimensionality reduction techniques which were applied on Named Entity Recognition task. Based on their analysis, they conclude that, Named Entity Recognition accuracy was poor for these languages. Performance of the classifier can be improved by dimensionality reduction [9].

Pati P et al. (2008) proposed algorithm for multi-script identification at the word level, they had started with a bi-script scenario which was extended to eleven-script scenarios. They used Support Vector Machine (SVM), Nearest Neighbor, and Linear Discriminate to evaluate Gabor and discrete cosine transforms features. They obtained accuracy of 98 % for up to tri-script cases, afterward they got 89 % accuracy [10].

Gupta V (2013) He had applied hybrid algorithm for Hindi and Punjabi language to summarize multilingual document. In proposed algorithm he had covered all most important features required for summarizing multilingual documents written in Hindi and Panjabi language and these features are: common part-of-speech tags like verb and noun, sentiment words like negative key word, position and key phrases, and named entities extraction. To identify weight of theses futures, he applied mathematical regression after calculating score of each features for each sentence. He got F-Score value of 92.56 % after doing experiment on 30 documents written in Hindi-Punjabi [11].

Hangarge M and Dhandra B (2008) they proposed a technique to identify Indian languages written in scanned version document based on morphological transformation features and its shape. They applied this technique on major Indian languages: Indian national language Hindi, old language Sanskrit and other two languages, and state languages Marathi, Bengali, and Assamese. They have created 500 blocks which contain more than two lines for each selected language. To decompose this blocks morphological transformation was used, after that they used KNN classifier and binary decision tree to classify these blocs. According to authors, this technique is quite different from other available technique for non-Indian language and they reported results were encouraging [12].

Padma M and Vijaya P (2010) they have proposed a method for language identification at the word level from trilingual document prepared using Hindi, English, and Kannada languages. The proposed method was trained by learning distinguish features of each language. After that they applied binary tree classifier to classify multilingual content. They obtained accuracy of 98.50 % for manually created database and average accuracy was found by 98.80 % [13].

3 Proposed Methodology

Based on the literature review and analysis of the tools available for Language Identification, we found that all researchers had used n-gram and other algorithm to identify particular language from multilingual document. We also analyzed that, these tools and methods cannot work for content separation. Existing work could not give proper and right output in case of mixed texts (for instance, “AअAઅAஅA”) appears in single sentence of multilingual document.

But none of the researcher has used Unicode Transformation Format for Language Identification purpose. In our proposed methodology, we have used UTF-8 for language identification. Each character of each language written in multilingual document or in a webpage could be identifying by its unique Unicode value. In order to design a methodology for Indian languages, we created a list of few Indian languages with their range of Unicode value. This list is presented in Table 2, Unicode range is also covered vowel, consonant, reserved language specific characters, digit,s and various sign used in particular language [3].

Table 2 Unicode range for Indian Languages

Sr. No.	Indian languages	Unicode range
1	Gujarati	0A80–0AFF
2	Panjabi	0A00–0A7F
3	Tamil	0B80–0BFF
4	Oriya	0B00–0B7F
5	Telugu	0C00–0C7F
6	Kannada	0C80–0CFF
7	Malayalam	0D00–0D7F
8	Bengali and Assamese	0980–09FF
9	Kaithi	11080–110CF
10	Devanagari, Hindi, Marathi, Sindhi, Nepali and Sanskrit	0900–097F

Figure 1 shows diagrammatic representation of methodology and how to implement the proposed methodology for Indian languages.

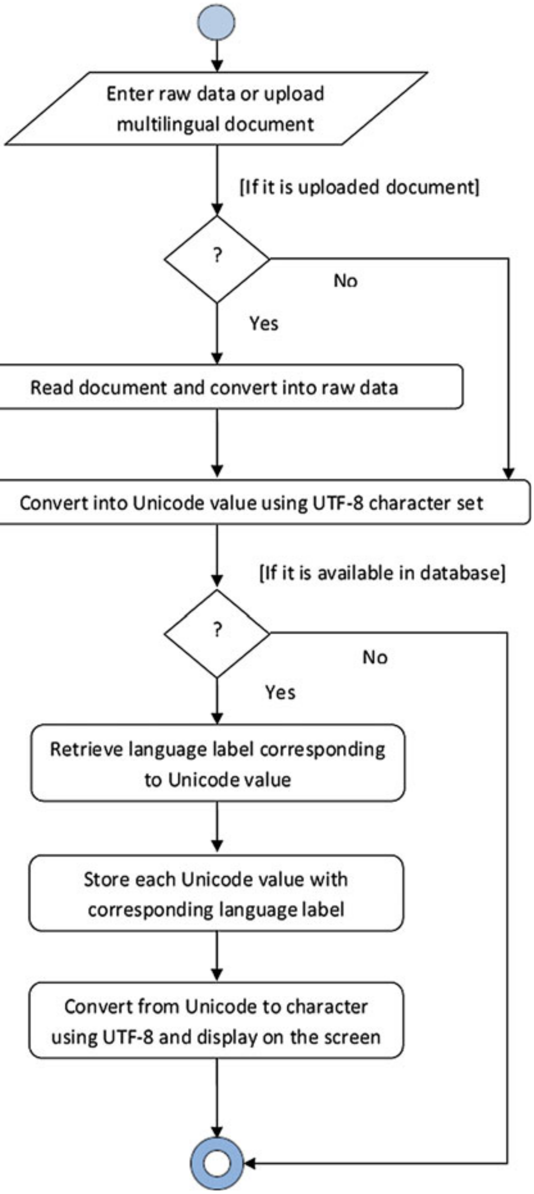
3.1 Advantages

- This method can be applied for mixed texts that appear in single world or sentence (for instance, "AअAઅAஔA").
- The proposed methodology is independent of font family of multilingual documents.
- It is also possible to implement this methodology in all most web technology other than JSP.
- It is free from the training phase.
- It can be extended for other language(s) by adding Unicode value in database.

3.2 Disadvantages

- It will lose the accuracy when multilingual document contain languages which has similar Unicode value, for instance languages Hindi, Marathi, and Devanagari (Table 2, Sr. No. 10).
- This methodology cannot be applied on scanned version of document.
- Loss of the accuracy in occurrence of mathematical sign, symbol, and special character appears in document.
- The proposed methodology losses the accuracy when corrupted text present in document.

Fig. 1 Flow of methodology


Enter raw data or upload multilingual document
[If it is uploaded document]
?
No
Yes
Read document and convert into raw data
Convert into Unicode value using UTF-8 character set
[If it is available in database]
?
No
Yes
Retrieve language label corresponding to Unicode value
Store each Unicode value with corresponding language label
Convert from Unicode to character using UTF-8 and display on the screen

4 Experimental Results and Evaluation

We have described our methodology in this section; we constructed a matrix that contains all possible Indian language with their Unicode values of each character of each Indian language.

4.1 Tools and Technology

We have used Java Server Pages to implement our proposed methodology; other software and tools are used [8, 14]:

- MyEclipse IDE (Editor)
- JDK 1.7 (development platform)
- MYSQL (database to store Unicode value of each character of each Indian language) [15]
- Google input (Enter data at run time for live experiment)
- Google Chrome (web browser)
- JSP 2.0 (to write a script)
- Tomcat Server (to execute JSP script) [16]

4.2 Languages and Data Sets

The collection consists of a corpus of texts collected randomly from the web for 16 different Indian languages: Gujarati, Hindi (extended devanagari), Punjabi (Gurmukhi), Bengali, Tamil, Telugu, Kannada, Marathi, Malayalam, Kashmiri, Assamese, Oriya, Kaithi, Sindhi, Nepali, and Sanskrit. After that, we had mixed the content written in different Indian languages and prepared 822 documents for experiment purpose. Each document contains at least five Indian languages with more than 800 words. We had also used Google input tool for live experiment to enter mixed content at run time through different users.

4.3 Results

We have done experiment on 822 different documents in which 522 prepared in Text file format and 300 in PDF (Portable Document Format). Each document containing at least five Indian Languages and more than 800 words. The documents belonged to different categories such as news, sports, education, politics, etc. We had collected the corpus texts randomly from web.

We achieved average accuracy of 99.63 % for entire system in which accuracy obtain 99.98 % from text file format and 99.28 % from PDF format. Text file format losing average accuracy by 0.02 %, because of conjunctions appear in documents written in some Indian languages like Gujarat and Hindi. Sometime overwritten conjunctions cannot read by stream classes and such character get skipped by the system.

We have randomly selected four records from obtained result of entire system which is presented in Table 3. After analyzing the result for entire system, we found

Table 3 Results

Sr. No.	Indian languages	Total number of words	Word based accuracy (%)		
		Pdf file	*Text file*	*Pdf file*	*Text file*
1	Gujarati, Oriya, Tamil, Panjabi and Bengali	849	920	99.88	100
2	Panjabi, Hindi, Malayalam, Kannada and Gujarat	822	811	99.76	100
3	Marathi, Assamesem Bengali, Tamil and Gujarat	802	825	99.63	99.88
4	Hindi, Marathi, Devanagari, Gujarati and Panjabi	840	902	63.76	64.88

that text file accuracy was more than that of PDF file by 0.70 %. The reason of getting loss of accuracy in PDF file was corrupted text (character get overwritten at the time of PDF creation) appeared in portable documents which is not interpret by system and it will skip it.

5 Conclusion and Future Work

We have used 8-bit Unicode value for automatic Indian language identification and content separation from multilingual documents. The obtained results with an execution of 522 Text file document, we achieved accuracy of 99.98 % and for the PDF accuracy found 99.28 % with an execution of 300 documents. The accuracy of text files is more than PDF files by 0.70 %. The result showing that, proposed methodology can be applied for document classification and a number of applications such as POS-Tagging, information retrieval, search engine, and machine translation for Indian languages. In future, we will apply this proposed methodology in document classification for Indian language.

References

1. "Department of Electronics & Information Technology, India", *Indian Language Technology Proliferation and Deployment Center* [Online]. Available: http://www.tdil-dc.in/index.php?option=com_up-download&view=publications&lang=en [May 10, 2015].
2. "Ministry of Communication & Information Technology, India", *Technology Development for Indian Languages* [Online]. Available: http://ildc.in/Gujarati/Gindex.aspx [May 10, 2015].
3. "The Unicode Consortium, USA", *The Unicode Standard* [Online]. Available: http://www.unicode.org/standard/standard.html [May 10, 2015].
4. Verma, V. K., & Khanna, N. (2013, April). Indian language identification using k-means clustering and support vector machine (SVM). In Engineering and Systems (SCES), 2013 Students Conference on (pp. 1–5). IEEE.

5. Anto, A., Sreekumar, K. T., Kumar, C. S., & Raj, P. C. (2014, December). Towards improving the performance of language identification system for Indian languages. In Computational Systems and Communications (ICCSC), 2014 First International Conference on (pp. 42–46). IEEE.
6. Yadav, P., & Kaur, S. (2013, November). Language identification and correction in corrupted texts of regional Indian languages. In Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O COCOSDA/ CASLRE), 2013 International Conference (pp. 1–5).IEEE.
7. Padma, M. C., Vijaya, P. A., & Nagabhushan, P. (2009, March). Language Identification from an Indian Multilingual Document Using Profile Features. In Computer and Automation Engineering, 2009. ICCAE'09. International Conference on (pp. 332–335). IEEE.
8. Chanda, S., Pal, U., & Terrades, O. R. (2009). Word-wise Thai and Roman script identification. ACM Transactions on Asian Language Information Processing (TALIP), 8(3), 11.
9. Saha, S. K., Mitra, P., & Sarkar, S. (2012). A comparative study on feature reduction approaches in Hindi and Bengali named entity recognition. Knowledge-Based Systems, 27, 322–332.
10. Pati, P. B., & Ramakrishnan, A. G. (2008). Word level multi-script identification. Pattern Recognition Letters, 29(9), 1218–1229. Chicago.
11. Gupta, V. (2013). Hybrid Algorithm for Multilingual Summarization of Hindi and Punjabi Documents. In Mining Intelligence and Knowledge Exploration (pp. 717–727). Springer International Publishing.
12. Hangarge, M., & Dhandra, B. V. (2008, July). Shape and Morphological Transformation Based Features for Language Identification in Indian Document Images. In Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on (pp. 1175–1180). IEEE.
13. Padma, M. C., & Vijaya, P. A. (2010). Word level identification of Kannada, Hindi and English scripts from a tri-lingual document. International Journal of Computational Vision and Robotics, 1(2), 218–235.
14. H. Marti and B. Larry, "Accessing Database with JDBC," in *Core Servlets and Java Server Pages volume 1*, 2nd ed. Pearson Education, 2008, ch.17, pp. 499–599.
15. "MySQL", *Unicode Support [Online]. Available:* http://dev.mysql.com/doc/refman/5.5/*en/charset-unicode.html [September 6, 2014].*
16. "The Apache Software Foundation", *Apache Tomcat 7* [Online]. Available: http://tomcat.apache.org/tomcat-7.0-doc/index.html [May 10, 2015].