Language Identifier for Languages of PakistanIncluding Arabic and Persian

Qaiser Abbas qaiser.abbas@uos.edu.pk
Department of Computer Science
University of Sargodha
Sargodha, 40100, Pakistan

M.S. Ahmad saeed@uos.edu.pk
Department of Computer Science
University of Sargodha
Sargodha, 40100, Pakistan

Sadia Niazi sadia_niazi2003@yahoo.com
Department of Pschology
University of Sargodha
Sargodha, 40100, Pakistan

Abstract

Language recognizer/identifier/guesser is the basic application used by humans to identify the language of a text document. It takes simply a file as input and after processing its text, decides the language of text document with precision using LIJ-I, LIJ-II and LIJ-III. LIJ-I results in poor accuracy and strengthen with the use of LIJ-II which is further boosted towards a higher level of accuracy with the use of LIJ-III. It also helps in calculating the probability of digrams and the average percentages of accuracy. LIJ-I considers the complete character sets of each language while the LIJ-II considers only the difference. A JAVA based language recognizer is developed and presented in this paper in detail.

Keywords:HAIL, LIJ, Di-grams, Identifier, Probabilistic.

1. LITERATURE REVIEW

Many techniques were adopted by the computational linguists to identify the language of text e.g. dictionary building, Markov models, tri-gram frequency vectors, and n-gram based text categorization, etc. These techniques are capable of achieving high degree of accuracy, large amount of memory space, and optimum time of processing. Among these, HAIL (Hardware Accelerated Identification of Language) [1] is one, based on hardware. This algorithm was designed on FPX (Field Programmable port Extender) platform [6]. It can detect four languages in a same document and generally, it can recognize 255 languages. The most important factor is the speed of processing; it can process data with a speed of 2.488 gigabits/second with accuracy. It uses N-gram of any length to detect language of a document [7] & [9]. This N-gram is further mapped with a hash on memory. N-gram extraction can be utilized by this system if the amount of the memory available is small e.g. one memory access for two, three and four characters used in HAIL. Since the HAIL identifies maximum four languages in a same document, for this purpose, it uses trend register to count the N-gram for respective languages. Three N-grams is optimal situation for this system, however, it detects beyond the limits up to four languages in a same document. The architecture of the HAIL system consists of eleven components which worked

together to perform the language identification on the network [8]. It contains tetra gram generator which process bytes from the TCP, it converts the ASCII characters into their respective uppercase letters and compressed them into 5-bit format. The stream then transferred to shift register and finally circuits extracts tetra grams [1]. SRAM reader uses theses tetra grams as addresses into SRAM dictionary and fetch up 8-bit language identifier from the address which is further used by count and score module to find the trends of the languages. The data is sent with its primary language to TCP re-serialization which is finally sent to report generator. HAIL records the addresses and counters and the report generator formats the data into UDP packet and transferred it to PC. SRAM programmer decodes the data in UDP format and used to program the dictionaries. HAIL achieves accuracy of 99 percent for documents containing one hundred words and its accuracy rate increases up to 99.95% if the file size increases.

A web search engine for a specific language also uses language identification algorithms e.g. indexing the Indonesian web. It searches only Indonesian (Bahasa Indonesian) web pages among all kind web pages written in other languages like English, Arabic, French, etc. Mainly, it has two objectives, one is to design search engine and second is to identify Indonesian language. I will discuss the language identification part only to focus on our objective. The methodology adopted in Indonesian language identification algorithm is based on to distinguish between Indonesian and non Indonesian languages. It learns from positive example only by devising the algorithm on frequency of tri-grams in Indonesian words [2] & [9]. The algorithm achieves a performance of 94% recall and 88% precision. As an experiment, 9 Indonesian documents were applied on algorithm. Performance measured after each of 10 iterations with a set of 24 documents containing 12 Indonesian and 12 English documents were applied to the algorithm. Further, after iteration, the performance is measured against a reference set containing 17 Indonesian, 4 English, 1 Malay, 1 Tagalog and 1 German document.

Another model for language identification is built for JAVA client/server platform. It uses the same methodology of N-gram with some additional features of labeling documents and text fragments. The JAVA programming language, portable virtual machine, web infrastructure and document resource protocol provides widely deployed platform for Natural Language Processing applications [3]. A probabilistic or profile based on character N-gram method is used for each language in classification set. After that a classification of unknown string with respect the model is performed to generate that string. Some issues like matching of input character with profile character set, documents on the web are insufficient to identify the language are removed by providing UNICODE support. Character co-occurrence problem is supported up to 5 characters in length in this model. Good Turing and conditional probabilities used by Dunning are explored in this experiment. The issue which is removed in this model is the low frequency items. It is experimented with low frequency items by using singleton in which N-grams are appeared only once in the training data. It leads to trade model size against accuracy but it is surprised to see the reasonable performance is remained even after filtering of singletons, even N-grams and even three times in training set. This model works extremely well in client web browser, document server and on proxy web server due to Java Virtual Machine. A *Frequency Table Class* written in Java is used for language labeling. *Main( )* routine is used for language profile creation for training data. The Frequency Table Class contains methods for saving and loading the profiles to disk and for scoring strings in profiles. Moreover, additional classes are used for client environment and for proxy HTTP server such as an applet and servlets respectively. This character N-gram language labeling algorithm is successfully used in Java Client Side Environment, offline document management system and in HTTP proxy server for NLP applications.

It has been accessed after viewing literature in the area that there are basically two different approaches. One is to create a list of letters/characters [15] for any language and then the match these letters with the letters of the document and second is to create a list of short words/strings [14] on which the model perform pattern matching to inform about the language used in the document. It is pertinent to note that many other approaches have also been adopted but these

approaches are the mixtures of these two basic strategies and are very complex in nature. Automated Language Processing System at USA produces a lot of work in natural language processing e.g. translators like ASK and TransMatic. The architecture of these can be seen in [4]. They designed a language identifier in 1987 based on ASK and TransMatic logic. It takes a buffer of text from the user environment and returns a buffer of information containing language of the text given. The language identifier is based on mathematical source language models uses cryptanalysis and probabilistic approach on character occurrence. It also stores a corpus of many languages to identify the language. Its working model is very simple; it takes corpus of any language in the memory, counts the total number of two letter sequence, counts the total number of occurrence of each distinct two letter sequence and finally divide the total number of occurrence of distinct two letter sequence with the total number of two letter sequences. This gives the probability of occurrence for a particular two letter word. Further detail regarding probability counting of digrams can be seen in [4].

2. DESIGN

The language recognizer/identifier for Arabic, Persian and Languages of Pakistan is difficult task which is still in progress. However, language recognizer for English and European languages can be found around the web world easily. Various techniques for building language identifier was already discussed in the literature review but here in this part, the design of my language recognizer is discussed . The selection of Java as tool for building LIJ (language recognizer in Java) depends upon flexibility and huge support for UNICODE files.

The design contains three levels to identify the language of a given input file to the LIJ model. In the first level, the model contains a record of character sets of seven languages including Urdu, Punjabi, Balochi, Sindhi, Pashto, Arabic, and Persian. The LIJ-I (Model Level I) takes some input file and after reading, matching is performed character by character with the character sets of languages and finally returns the frequency of each language's characters. The LIJ-I decides the language by calculating this simple concept given below in equation (1).

$$RL = Set\_Max\_f\ (\ Lang\ Chars'\ Set\ of\ freqs'\ ) \qquad (1)$$

The design of LIJ-I is shown below in figure -1.The BRU is a buffering technique available in Java. These buffers are used for each language and input file respectively. The results of LIJ-I for languages of Pakistan and for Arabic and Persian are given in table-1 while the size of input file is 3500 words which includes portion for each language contains 500 words respectively.

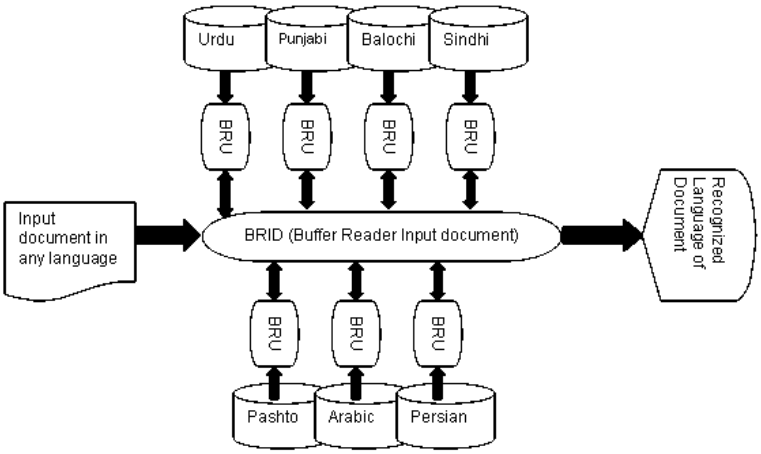

FIGURE1: Design of LIJ-I

Similarly in the second level LIJ-II, the system has already an input file as given in LIJ-I. The LIJ-II has record of common characters from the character sets of seven languages. Moreover, it has also records for each language that contain specifically different characters from their respective character sets of languages. The LIJ-II first read the input document in a buffer and similarly the specific different characters of each language into their respective buffers. The LIJ-II starts matching the character of input document with the specific different (SD) records and counts the occurrence of SD characters of each language. Based on this occurrence of the SD characters of a language, it decides the language of document. The concept is given in equation (2) below.

$$RL = Set\_Max\_f(Lang\ Chars'Set\ of\ SD\ freqs') \quad (2)$$

It works with almost a little bit improvement in accuracy as shown in the table 2, then we add up the results of LIJ-I and LIJ-II as depicted in table 3 in the next section. The design of LIJ-II is shown in figure 2 below.

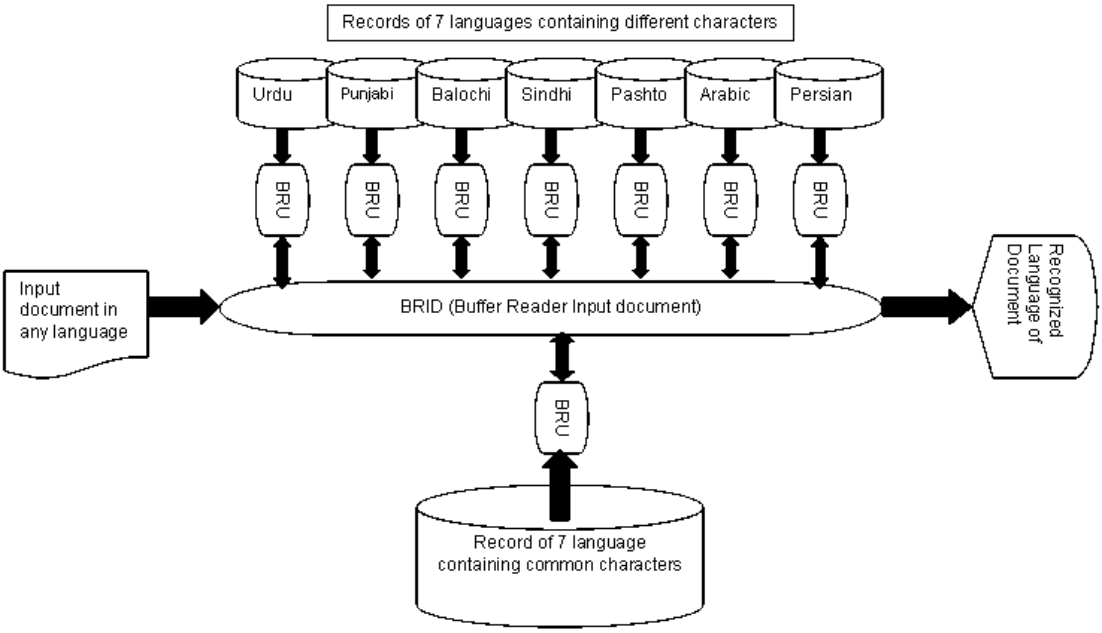

FIGURE 2: Design of LIJ-II

The third level of language recognizer LIJ-III is implemented on the logic presented by Kenneth R. Beesley in [4]. At first, corpus based probability of occurrence is found by converting text in a corpus into two letters' partition known as digrams. After conversion, counts the total numbers of digrams and then counts the occurrence of each different digram in the corpus and divide its occurrence with the total number of digrams in corpus. This gives us the probability of occurrences which is stored in LIJ-III. Now LIJ-III deals with the input file. It converts words into digrams first and starts with each particular digram and the value of the probability of occurrence of each digram is stored. After that these occurrence of digrams are multiplied with respect to a particular language. The language with highest probability is the decision made by the LIJ-III. As an example suppose that a Urdu corpus contained 20000 total digrams (two letter words) and 1000 distinct digrams and digram 'مق'(*muq*) of word 'مقام' (*muqam* means location) appears 15 times therefore the probability of occurrence of 'مق' (*muq*) digram is calculated as P_{Urdu}(مق) = 15 / 20000 = 0.00075. The whole computation is given in the following equation 3 in which D is a digram and TD is the total number of digrams in a given corpus.

$$P(D_{lang}) = \prod_{i=1}^{n} count(D_i)/TD_{corpus} \quad (3)$$

Beesley [4] focused to identify a single main language in a given document but we modified his approach to not only identify the main language of the document probabilistically but also the share of each language in a given document. The design of the LIJ-III is shown in figure 3. It is pertinent to note that the LIJ-III generates digrams of the whole input UTF-8 file.

The one final step after the performance of these three levels discussed, the model LIJ generates the final decision on the average percentage of these levels LIJ-I & LIJ-II (Table 3) and LIJ-III (Table 4). The language with maximum percentage is finalized as the main language of the document with percentage weights of other languages.

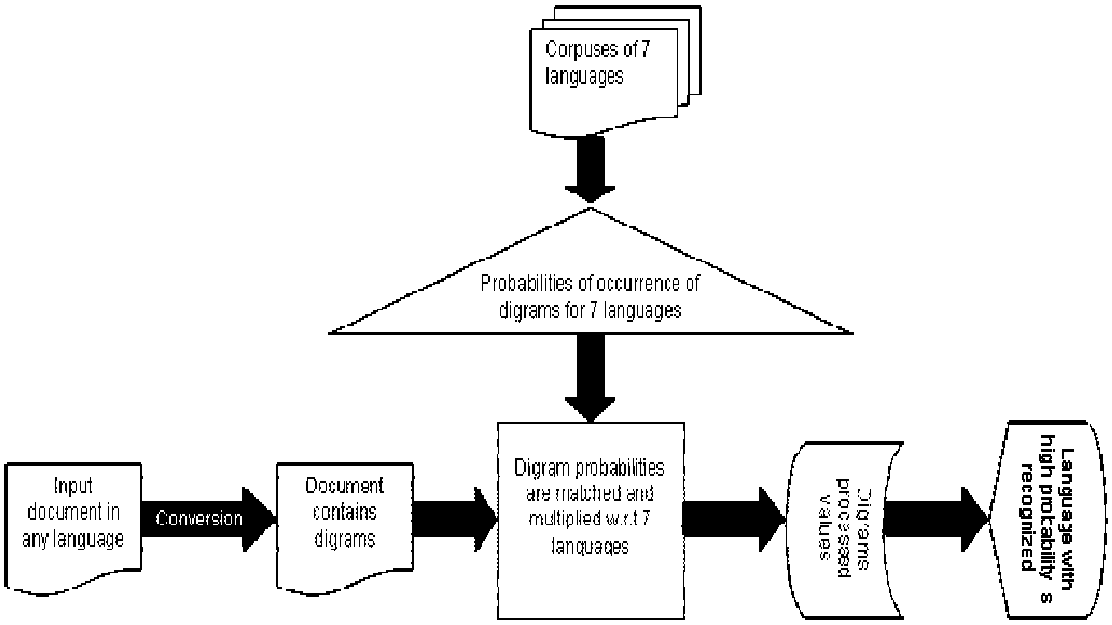

FIGURE3: Design of LIJ-III

3. RESULTS

As the LIJ-I contain the sets of character alphabets for each language and an input file contains data of the languages of Pakistan including the Arabic and the Persian. The file size is of about 3500 words with an estimate of 500 words for each language. The language with highest frequency of characters becomes the main language of input file by the LIJ-I as shown in table-1 in case of Arabic.

Language	Correct	Wrong	%age
Urdu	2499	1001	71.4
Punjabi	2184	1316	62.4
Balochi	2268	1232	64.8
Sindhi	2415	1085	69.0
Pashto	2345	1155	67.0
Arabic	**2555**	**945**	**73.0**
Persian	2177	1323	62.2

Table1: Result of LIJ-I

The entries in the 'wrong' column of table-1 indicate that there is some unmatched criterion which results in poor accuracy and more importantly predicting the behavior of fertile languages.

In LIJ-II, our approach is almost the contrast of LIJ-I, we focused on the sets of specific different characters in each language and matched with the input file as mentioned earlier. The results of LIJ-II are shown in the following table 2 .

Language	Correct	Wrong	%age
Urdu	**720**	**2780**	**20.6**
Punjabi	542	2958	15.5
Balochi	451	3049	12.9
Sindhi	489	3011	14.0
Pashto	521	2979	14.9
Arabic	624	2876	17.8
Persian	376	3124	10.7

TABLE2: Result of LIJ-II

The percentage of accuracy is improved when we add up the results of LIJ-I and LIJ-II. The reason is LIJ-II's approach uncovers the ambiguity lying in LIJ-I with complete character sets and each character set contain common and different characters in a unit. So, we can say simply that LIJ-II is an attempt to uncover the hidden share of accuracy. Moreover, it is pertinent to note that the preference of the Arabic is converted to the Urdu language after addition. This is the effect that mostly languages of Pakistan like Punjabi, Balochi, Sindhi, etc. shares a lot of characters with the Urdu language and hence contribute to increase its share in the results given in table-3.

Lang	LIJ-I %	LIJ-II %	Acc. %age
Urdu	**71.4**	**20.6**	**92.0**
Punjabi	62.4	15.5	77.9
Balochi	64.8	12.9	77.7
Sindhi	69.0	14.0	83.0
Pashto	67.0	14.9	81.9
Arabic	73.0	17.8	90.8
Persian	62.2	10.7	72.9

TABLE3: Accumulative Result of LIJ-I and LIJ-II

The design of LIJ-III is discussed in the previous section and the results obtained through this approach are shown in Table 4. The main language of the given test input file is highlighted as a bold while the percentages of other languages are also calculated.

Language	Correct	Wrong	%age
Urdu	**2999**	**501**	**85.7**
Punjabi	2714	786	74.5
Balochi	2486	1014	71.0
Sindhi	2045	1455	58.4
Pashto	2163	1337	61.8
Arabic	2513	987	71.8
Persian	2611	889	74.6

TABLE4: Result of LIJ-III

The results show a great improvement in case of the Urdu, Punjabi, Balochi, Arabic and Persian language which clearly depicts a morphological similarity in these languages while the results of Sindhi and Pashto are at a difference from other language gives information about their dissimilarity in morphological structure. The Corpus used in this experiment for digram probability calculation contained 35,000 words approximately and the development data and the test data contain 3500 words in each. Finally, the average percentage is calculated through adding Table 3 and Table 4. The results are shown in Table 5 below.

Language	Acc. %age (Table 3)	%age (Table 4)	Avg. %age
Urdu	**92.0**	**85.7**	**88.9**
Punjabi	77.9	74.5	76.2
Balochi	77.7	71.0	74.4
Sindhi	83.0	58.4	70.7
Pashto	81.9	61.8	71.9
Arabic	90.8	71.8	81.3
Persian	72.9	74.6	73.8

TABLE 5: Average Percentage of Languages

This average percentage gives a lot of improvements as per gold standard with some ambiguities in case of the Arabic language [10]. However, when the language of input file is mainly the Arabic, it gives correct decision. So, this ambiguity remains no ambiguity at all.

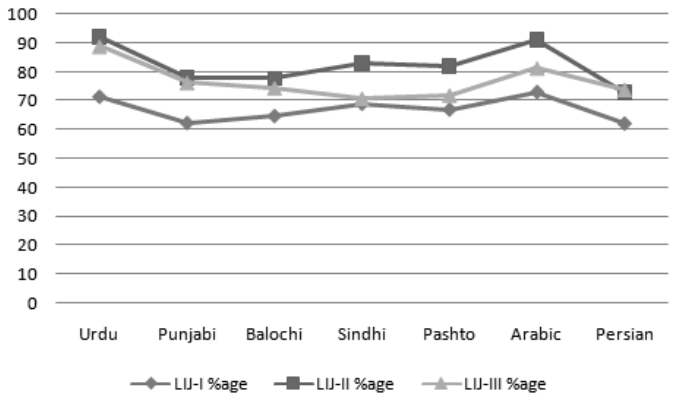

FIGURE 4: Improvement Comparison of LIJ's

The Figure 4 clearly depicts that the Urdu, Punjabi and Persian improves the most in our model and gets the first ranking position in the language set while the Balochi language remains on second ranking position, The Arabic is on 3rd , Pashto is on 4th and Sindhi is on 5th ranking position. This gives us a clue that our model handles the languages extracted from the mother language (The Arabic) in quite a good way.

We could not compare our results of languages of Pakistan with other work done for English and European languages because of the nature of languages and no any work regarding the identification of languages of Pakistan is existed in literature. However, in general, irrespective of technology or method used, HAIL [1] approach in case of Arabic language is more fruitful than our approach with 94% of accuracy having 500 words in training set which is similar to our size of corpus for this language. We obtained maximum accuracy 90.8% as depicted in the above line chart. Similarly, word fragment based work in [15] has concluded 96.6% accuracy in case of Arabic, however, it was mentioned that this accuracy percentage held for Urdu and Pashto too

but no such results were depicted in their work. It is also claimed that all the six languages including Arabic, Persian, Urdu, Pashto, Uighur and Kurdish have the same accuracy percentage 96.6% which is no doubt an ambiguous statement.

4. DISSCUSSION AND ISSUES

During this development, a lot of problems are being faced, majority issues were solved but some of issues are very much complex in their nature while the others are unknown to me. The detail is as under by points:

i. Limited literature with respect to languages of Pakistan is no doubt a big issue.
ii. Buffer reader/writer related problems in code were not expected but they arrived and their removal was a hectic job.
iii. Due to fertility of languages, the LIJ-I fails to predict Persian, Punjabi and Balochi accurately.
iv. Similarly Persian language has a very close resemblance with Pashto, so occurrence of many letters in both the languages is same causing LIJ-I to fail in detecting it properly, in this situation LIJ-II plays an important role and the decision is made on the accumulative percentage of LIJ-I and LIJ-II.
v. Punjabi language lacks in data availability on the web or in the form of digital corpus.
vi. The languages discussed in this paper contain no space distinction for words. So, space is inserted between words in a corpus before processing using Joiners, non-Joiners and manual insertion method.
vii. The corpus for languages has been collected from different websites available on internet. The corpus for Urdu, Arabic, Persian language has been mostly collected from the Daily Jang News Paper, British Broadcasting Corporation and others[1]. The Pashto language data is obtained mostly from afghan website[2]. Similarly Balochi language is obtained from only a single website even tried to search a lot but in vain[3]. Sindhi language is spreading due to Karachi city because a lot of efficient people are there and working on this language. Many website regarding Sindhi can be seen in UNICODE form, among them some are used for collecting the data[4] and finally the most important language of Punjab province which is far better than other provinces but unfortunately the people are not interested in doing work regarding Punjabi. Not a single website is viewed by me to get data for Punjabi language. However, research papers/articles are used to get some of its data [5] & [11]. There are a lot of issues regarding computational resources in Pakistan briefly described in [13].
viii. The five languages of Pakistan mentioned in the paper and Persian has its roots in Arabic and also has ambiguities in their respective character sets. Due to which a high accuracy in language identification is really a hard problem. All the languages shares a common character set whose size is more than half of their respective characters. The Urdu language has lot of ambiguities in its character set and collating sequence [12].

[1] Urdu Daily Jang News Pakistan athttp://www.jang.com.pk/ , for Arabic Newstin News and People at http://www.newstin.ae/sim/ar/76065072/ar-010-000224316, and شبكة أبناء ليبيا , http://libyasons.com/vb/showthread.php?t=59428 and الأخبارhttp://www.aljazeera.net/NR/EXERES/6614C6F0-E7FB-41E0-AD2A-04BF526C416F.htm

[2] For Pashto: Bakhtar News Agency athttp://www.bakhtarnews.com.af/ and http://www.tolafghan.com/paktia_pa_dag_ke , British Broadcasting Corporation for Pashto: http://www.bbc.co.uk/pashto and http://www.shahadatnews.com/

[3] For Balochi: BalochiZuban-o-Adab-e-Dewan: http://www.baask.co.cc/

[4] For Sindhi: http://www.sindhilife.com, http://www.sarangaa.com and http://www.sindhiadabiboard.org

5. CONSLUSION

The LIJ developed for the languages of Pakistan including the Arabic and the Persian is the first one in its own nature. The respective accuracy percentage of each language is not obtained as expected but despite of all this, it is the first language identifier which has accuracy percentage at this high level for languages of Pakistan. It used a very simple and probabilistic approach to give a final decision about the language of the document or input file. The most and prominent hurdle that such work has not been initiated in the Past is the non availability of the computational resources, and non standardization of the computational resources available even the present is suffering too.

6. REFERENCES

[1] Charles M. Kastner, G. Adam Covington, Andrew A. Levine, John W. Lockwood, "*HAIL: A HARDWARE-ACCELERATED ALGORITHM FOR LANGUAGE IDENTIFICATION"*, 15th Annual conference on Field Programmable Logic and Applications (FPL), USA, 2005.

[2] V. Berlian, S.N. Vega, and S. Bressan, "*Indexing the Indonesian web: Language identification and miscellaneous issues*", In the *Tenth International World Wide Web Conference*, Hong Kong, 2001.

[3] Gary Adams and Philip Resnik. "*A language identification application built on the Java client-server platform*". In Jill Burstein and Claudia Leacock, editors, *From Research to Commercial Applications: Making NLP Work in Practice*, pages 43--47. Association for Computational Linguistics, 1997.

[4] K. R. Beesley. "*Language identifier: A computer program for automatic natural-language identification on on-line text*". In *Proceedings of the 29th Annual Conference of the American Translators Association*, pages 47—54, USA, 1988.

[5] Tejinder Singh Saini1 and Gurpreet Singh Lehal2, "*Shahmukhi to Gurmukhi Transliteration System: A Corpus based Approach*", Research in Computing Science (Mexico), Vol-33, Pages 151-162. USA, 2008.

[6] J. Lockwood, J. Turner, and D. Taylor, "*Field Programmable Port Extender (FPX) for Distributed Routing and Queuing*" in ACM International Symposium on Field Programmable Gate Arrays (FPGA), 2000.

[7] Bashir Ahmed, Sung-Hyuk Cha, and Charles Tappert. "*Language identification from text using n-gram based cumulative frequency addition*". In *Proc. of CSIS Research Day*, pages 12.1–12.8, Pace University, NY, 2004.

[8] D. Schuehler and J. Lockwood, "*A Modular System for FPGA-based TCP Flow Processing in High-Speed Network,*" in 14th International Conference on Field Programmable Logic and Applications (FPL), Antwerp, Belgium, pp. 301–310, 2004.

[9] Cavnar, William B., Trenkle, M. "*N-gram based text categorization",* InProceedings of the third Annual Symposium on Document Analysis and Information Retrieval, pp161-169, 1994.

[10] Hussain, S., Karamat N., Mansoor, A. "*Arabic Script Internationalized Domain Names*", In the Proceedings of the CIIT Workshop on Research in Computing, CWRC'08, CIIT Lahore, Pakistan, 2008.

[11] M.G.A. Malik, "*Towards Unicode Compatible Punjabi Character Set"*, Proceeding of 27th Internationalization and Unicode Conference, Berlin, Germany, 2005,.

[12] Hussain, S. "*Urdu Collation Sequence*", In the Proceedings of the IEEE International Multi-Topic Conference, Islamabad, Pakistan, 2003.

[13] Hussain, S. "*Computational Linguistics in Pakistan: Issues and Proposals*", In the Proceedings of EACL (Workshop in Computational Linguistics for Languages of South Asia), Hungary, 2003.

[14] C. Kruengkrai, P. Srichaivattana, V. Sornlertlamvanich, and H. Isahara. "*Language identification based on string kernels*". In *Proceedings of the 5th International Symposium on Communications and Information Technologies,* 2005.

[15] Hisham El-Shishiny, Alexander Troussov, "*Word Fragments Based Arabic Language Identification*", NEMLAR, Arabic language Resources and Tools Conference, Cairo, Egypt, 2004.