International Journal of Computational Linguistics (IJCL)

ISSN : 2180-1266

Volume 1, Issue 3

Number of issues per year: 6

International Journal of Computational Linguistics (IJCL)

Volume 1, Issue 3, 2010

Edited By
Computer Science Journals
www.cscjournals.org

International Journal of Computational Linguistics (IJCL)

Book: 2010 Volume 1, Issue 3
Publishing Date: 20-12-2010
Proceedings
ISSN (Online): 2180-1266

IJCL Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Editorial Preface

The International Journal of Computational Linguistics (IJCL) is an effective medium for interchange of high quality theoretical and applied research in Computational Linguistics from theoretical research to application development. This is the thired issue of volume first of IJCL. The Journal is published bi-monthly, with papers being peer reviewed to high international standards. International Journal of Computational Linguistics (IJCL) publish papers that describe state of the art techniques, scientific research studies and results in computational linguistics in general but on theoretical linguistics, psycholinguistics, natural language processing, grammatical inference, machine learning and cognitive science computational models of linguistic theorizing: standard and enriched context free models, principles and parameters models, optimality theory and researchers working within the minimalist program, and other approaches.

IJCL give an opportunity to scientists, researchers, and vendors from different disciplines of Artificial Intelligence to share the ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development. This journal is helpful for the researchers and R&D engineers, scientists all those persons who are involve in Computational Linguistics.

Highly professional scholars give their efforts, valuable time, expertise and motivation to IJCL as Editorial board members. All submissions are evaluated by the International Editorial Board. The International Editorial Board ensures that significant developments in image processing from around the world are reflected in the IJCL publications.

IJCL editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Scribd, CiteSeerX Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCL. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJCL provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Computational Linguistics (IJCL)

Table of Content

Volume 1, Issue 3, December 2010

Pages

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).

$$RL = Set\_Max\_f\ (\ Lang\ Chars'\ Set\ of\ freqs') \tag{1}$$

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.

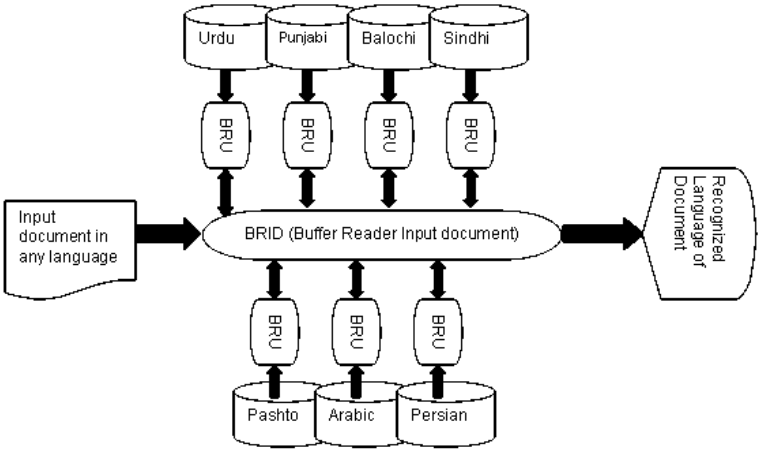
Urdu
Punjabi
Balochi
Sindhi
BRU
BRU
BRU
BRU
Input document in any language
BRID (Buffer Reader Input document)
Recognized Language of Document
BRU
BRU
BRU
Pashto
Arabic
Persian

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.

$$RL = Set\_Max\_f(Lang\ Char\ s'Set\ of\ SD\ freq\ s') \qquad (2)$$

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.

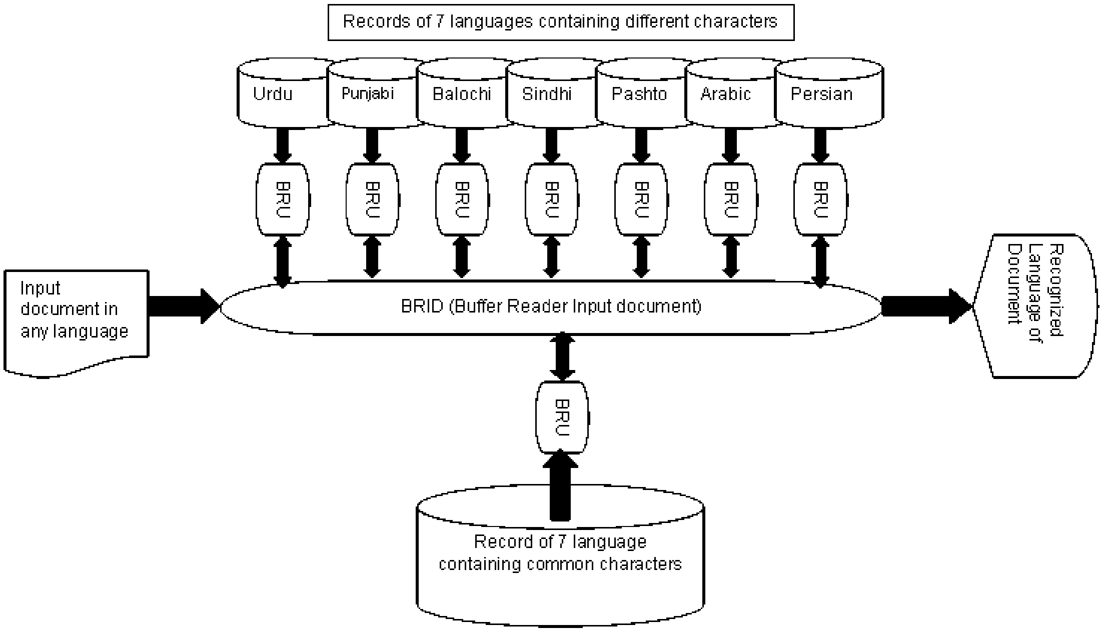
Records of 7 languages containing different characters
Urdu
Punjabi
Balochi
Sindhi
Pashto
Arabic
Persian
BRU
Input document in any language
BRID (Buffer Reader Input document)
Recognized Language of Document
Record of 7 language containing common characters

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.

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

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.

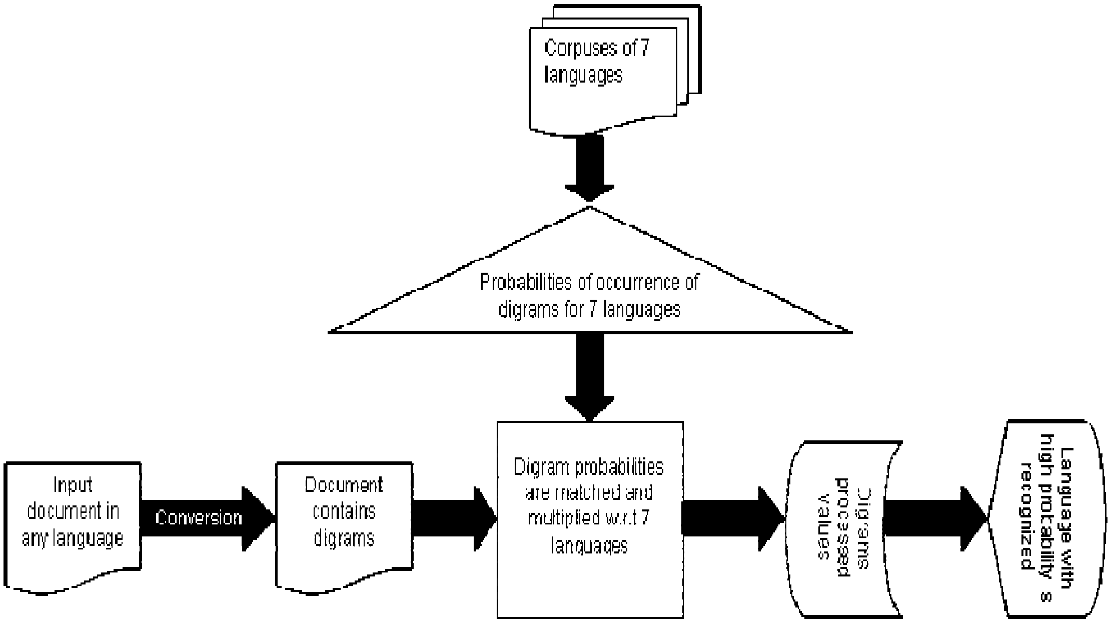
Corpuses of 7 languages
Probabilities of occurrence of digrams for 7 languages
Input document in any language
Conversion
Document contains digrams
Digram probabilities are matched and multiplied w.r.t 7 languages
Digrams processed values
Language with high probability s recognized

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.

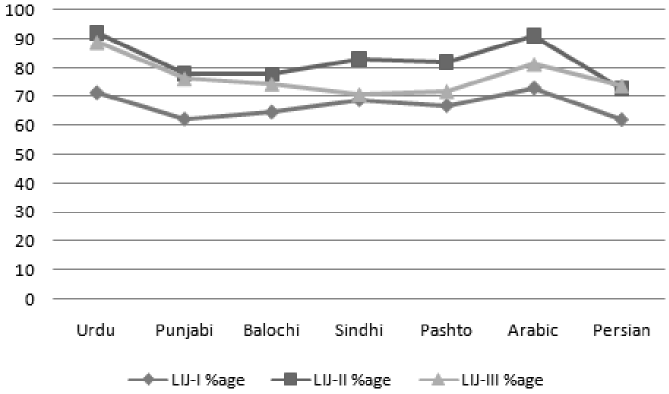

FIGURE 4: Improvement Comparison of LIJ's

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.

Named Entity Recognition for Telugu Using Conditional Random Field

G.V.S.Raju letter2raju@gmail.com
Professor of CSE Department
Indur institute of Engg.&Tech
Siddipet,A.P, India

B.Srinivasu srinivas_534@yahoo.com
Asso Professor of CSE Department
Indur institute of Engg.&Tech
Siddipet,A.P, India

S.VISWANADHA RAJU viswanadharajugriet@gmail.com
Professor of CSE Department
JNTUH College of Engineering
Jagityala , A.P, India

ALLAM BALARAM balaramallam@gmail.com
Asst professor of CSE Department
Indur institute of Engg.&Tech
Siddipet,A.P, India

Abstract

Named Entity (NE) recognition is a task in which proper nouns and numerical information are extracted from documents and are classified into predefined categories such as Person names, Organization names , Location names, miscellaneous(Date and others). It is a key technology of Information Extraction, Question Answering system, Machine Translations, Information Retrial etc. This paper reports about the development of a NER system for Telugu using Conditional Random field (CRF). Though this state of the art machine learning technique has been widely applied to NER in several well-studied languages, the use of this technique to Telugu languages is very new. The system makes use of the different contextual information of the words along with the variety of features that are helpful in predicting the four different named entities (NE) classes, such as Person name, Location name, Organization name, miscellaneous (Date and others).

Keywords: Named entity, Conditional Random field, NE, CRF, NER, named entity recognition,Telugu

1. INTRODUCTION

Named Entity (NE) recognition is an important tool in almost all natural language processing applications like Information Extraction (IE), Information retrieval and machine translation and Question answering system etc. The objective of NER is detect and classify each and every word or token in a text document into some predefined categories such as person name, location name, organization name, date and designation. Identification of named entity is a difficult task because named entities are open class expressions, i.e there is an infinite verities and new expressions are constantly being invited.

NER system has been developed for resources rich language like English is very high accuracies. But development of NER system for a resource poor language like Telugu is very challenging due to unavailability of proper resources.[5]English is resource-rich language containing lots of resources for NER and other NLP tasks. Some of the resources of English language can be used to develop NER system for a resource-poor language. Also English is used widely in many countries in the world. In India, although there are several regional languages like Telugu, Kannada, Tamil, Hindi etc.., English is widely used (also as subsidiary official language). Use of the Telugu languages in the web is very little compared to English and other Indian languages. So, there are a lot of resources on the web, which are helpful in Telugu language NLP tasks, but they are available in English. For example, we found several relevant name lists on the web which are useful in Telugu NER task, but these are in English. It is possible to use these English resources if a good transliteration system is available.

Transliteration is the practice of transcribing a word or text in one writing system into another. Technically most transliterations map the letters of the source script to letters pronounced similarly in the goal script. Direct transliteration from English to an Telugu language is a difficult task.

A large number of techniques have been developed to recognize named entities for different languages. Some of them are Rule based and others are Statistical techniques. The rule based approach uses the morphological and contextual evidence (Kim and Woodland,2000) of a natural language and consequently determines the named entities. This eventually leads to formation of some language specific rules for identifying named entities. The statistical techniques use large annotated data to train a model (Malouf, 2002) (like Hidden Markov Model) and subsequently examine it with the test data. Both the methods mentioned above require the efforts of a language expert. An appropriately large set of annotated data is yet to be made available for the Indian Languages. Consequently, the application of the statistical technique for Indian Languages is not very feasible. This paper deals with a CRF technique to recognize named entities of Telugu languages.

2. NER FOR INDIAN LANGUAGES

NLP research around the world has taken giant leaps in the last decade with the advent of effective machine learning algorithms and the creation of large annotated corpora for various languages. However, annotated corpora and other lexical resources have started appearing only very recently in India. Not much work has been done in NER in Indian languages in general and Telugu in particular. Here we include a brief survey.

In (Eqbal, 2006), a supervised learning system based on pattern directed shallow parsing has been used to identify the named entities in a Bengali corpus. Here the training corpus is initially tagged against different seed data sets and a lexical contextual pattern is generated for each tag. The entire training corpus is shallow parsed to identify the occurrence of these initial seed patterns. In a position where the seed pattern matches wholly or in part, the system predicts the boundary of a named entity and further patterns are generated through bootstrapping. Patterns that occur in the entire training corpus above a certain threshold frequency are considered as the final set of patterns learned from the training corpus.

In (Li and McCallum, 2003), the authors have used conditional random fields with feature induction to the Hindi NER task. The authors have identified those feature conjunctions that will significantly improve the performance. Features considered here include word features, character n-grams (n = 2,3,4), word prefix and suffix (length - 2,3,4) and 24 gazetteers.

3. SOME CASES IN TELUGU LANGUAGE NAMES

Some of the typical ambiguous cases in Telugu
variation of Named Entities:
వైయస్ రాజశేఖర్ రెడ్డి వైయస్, వై.యస్.ర్
vaiyas raajas`eekhar reDDi, vaiyas, vai.yas.r.
Y.S. Rajashakar Reddy, Y.S. Y.S.R
Ambiguity in NE type:
సత్యం (SatyaM) person Vs organizatio
తిరుపతి (Tirupati) person Vs location
Ambiguity with Common Noun:
బంగారు Gold (baMgaaru) person first Vs common noun
Appearance in various forms:
తెలుగుదేశంపార్టీ, టీడీపీ, తె.దే.పా
telugudees`aMpaarTii, TiiDiiPii, te.dee.paa
Telugu Desam party, T.D.P, Te.De.Pa
These are some examples which show the complexity of development of NER system .This paper is focused on development of NER for Telugu Language

4. APPROACHES FOR NER

Named entity recognition is a classification and identification problem but this is a kind of problem which requires features of the word at least in case of Telugu for proper identification of NEs. Widely used approaches for solving such problems are Statistical Machine learning Techniques, Rule Based System or hybrid approach. In Statistical techniques many approaches may be applied like Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM) and Conditional Random Field (CRF) [3], support vector machine(SVM). Sequence labeling problem can be solved very efficiently with the help of Markov Models. The conditional probabilistic characteristic of CRF and MEMM are very useful for development of NER system. Between both, MEMM is having bias labeling problem which makes it vulnerable for this research. CRF is flexible enough to capture many correlated features, including overlapping and non Independent features. Thus multiple features can be used in CRF more easily than in HMM.

All Machine Learning Techniques require a large relevant corpus which can pose a problem in case of Telugu and other Indian Languages because of unavailability of such corpus. The positive side of these techniques is being cost effective and requires less language expertise whereas Rule based system requires language expertise for crafting rules. A rule based system incurs huge cost and time.

5. DEVELOPMENT OF TAGGED DATA

News articles generally start with a headline and the body starts with location name, month and date. A seed list of location names is extracted from this. Seed lists of personal surnames, location names and organization names have also been developed. Lists of person suffixes such as “reDDi(reddy)”, “naayuDu(nayudu)” etc, location suffixes such as “baad”, “peeTa” “paTnaM” etc, and name context lists are maintained for tagging the corpus. It has been observed that whenever a context word (such as ”maMtri”) appears, then in many cases the following two words (consisting of a surname and person name) indicate a person name. This way we build a list of person names. After extensive experimentation over many iterations, a training data set of 30,000 words has been developed.

6. NOUN IDENTIFICATION

It is useful to recognize nouns and eliminate non-nouns. The Telugu morphological analyzer developed here has been used to obtain the categories. A stop word list including function words has been collected from existing dictionaries and stop words are removed. Words with less than three characters are unlikely to be nouns and so eliminated. Last word of a sentence is usually a verb(Telugu is verb final language, in ever sentence final word may be a verb) and is also eliminated. Digits are eliminated. Verbs are recognized based on a list of verb suffixes and eliminated. Telugu words normally end with a vowel and consonant ending words (laMDan(Landan), sTeeSan(station), meenejar (manager) etc.) are usually nouns. Existing dictionaries are also checked for the category. Using these features, a naive Bayes classifier is built using the available tool WEKA. Results are given in the tables 1 and 2.

	noun	not-noun
Precision	94.56	63.47
Recall	62.78	94.45
F-measure	76.25	75.38

TABLE 1: Noun Identification using Morphological Analyzer

	Test set-1		Test set-2	
	noun	not-noun	noun	not-noun
Precision	91.56	95.56	76.47	89.2
Recall	96.15	91.45	90.45	74.48
F-measure	95.1	93.67	83.28	81.15

TABLE 2:. Noun Identification using a naive Bayes Classifier

7. PROPOSED METHODOLOGY FOR NER

7.1 Labeling Sequential Data

The task of assigning label sequences to set of observation sequences arises in many fields, including speech recognition, computational linguistics. For example, consider the natural language processing task of labeling the words in a sentence with their corresponding part-of-speech (POS) tags. In this task, each word is labeled with the tag indicating it's appropriate part-of-speech resulting in annotated text such as :

<NPER>caMdrabaabu naayuDu <NLOC>raajoli graamamunu <V> saMdars`iMcaaru.

Chandrababu Naidu visited Rajoli village .

Labeling sentences in this way is a useful prepossessing step for higher NLP tasks: POS tags augment the information contained with in the words alone by explicitly indicating some of the structure inherent in the language.

One of the most common methods for performing such labeling and segmenting tasks is that of employing hidden Markov models (HMMs) or probabilistic finite-state automata to identify the most likely sequence of labels for the words in a given sentence. HMMs are a form of generative model, that defines a joint probability distribution p(X|Y) where X and Y are random variables respectively ranging over observation sequences and their corresponding label sequences. In order to define a joint distribution of this nature, generative models must enumerate all possible observation sequences a task which, for most domains,is intractable unless observation elements are represented as isolated units,independent from the other elements in an observation sequence. More precisely, the observation element at any given instant in time may only directly depend on the state, or label, at that time. This is an appropriate assumption for a few simple data sets, however most real-world observation sequences are best represented in terms of multiple interacting features and long-range dependencies between observation elements.

This representation issue is one of the most fundamental problems when labeling sequential data. Clearly, a model that supports tractable inference is necessary, however a model that represents the data without making unwarranted independence assumptions is also desirable. One way of satisfying both these criteria is to use a model that defines a conditional probability p(Y|X) over label sequences given a particular observation
sequence x, rather than a joint distribution over both label
and observation sequences. Conditional models are used to label a novel observation sequence x_* by selecting the label sequence y_* that maximizes the conditional probability p(y_*| x_*) The conditional nature of such models means that no effort is wasted on modeling the observations, and one is free from having to make unwarranted independence assumptions about these sequences; arbitrary attributes of the observation data may be captured by the model, without the modeler having to worry about how these attributes are related. Conditional random fields [lafferty](CRFs)are a probabilistic framework for labeling and segmenting sequential data, based on the conditional approach described in the previous paragraph. A CRF is a form of undirected graphical model that defines a single log-linear
distribution over label sequences given a particular observation sequence. The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference.[D. Pinto,F. Sha, lafferty]

7.2 Undirected Graphical Models

A conditional random field may be viewed as an undirected graphical model, or Markov random field, globally conditioned on X, the random variable representing the observation sequences. Formally, we define G = (V,E) to be an undirected graph such that there is a node v in V corresponding to each of the random variables representing an element Yv of {Y}. If each random variable Y{v} obeys the Markov property with respect to G, then (Y,X) is a conditional random field In theory the structure of graph G may be arbitrary, provided it represents the conditional independencies in the label sequences being modeled. However, when modeling sequences, the simplest and most common graph structure encountered is that in which the nodes corresponding to elements of Y form a simple first order chain as illustrated in figure 1.

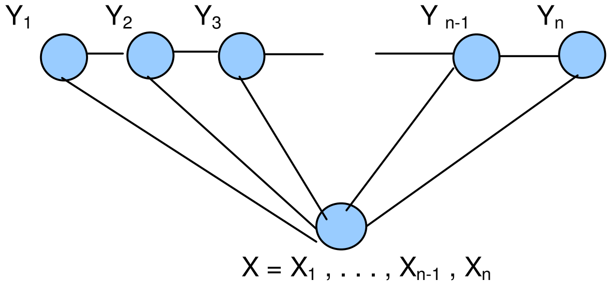

FIGURE 1

Let X is a random variable over data sequences to be labeled and Y is a random variable over corresponding label sequences. All components Y_i of Y are assumed to range over a finite label alphabet Y. For example, X might range over natural language sentences and Y may range over part-of-speech tagging of those sentences, with Y being set of possible part-of-speech tags.

Definition : Let G = (V,E) be a graph such that Y= Y_v, vεV, so that Y is indexed by vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X the random variables $Y_{v,}$ obey the Markov property with respect to graph : p(Y_v |X, Y_w, w≠ v)= p(Yv| X, Yw, w • v) where w • v means w and v are neighbors in G.
A CRF is a random field globally conditioned on the
observation X.

$$p(\mathrm{y}|\mathrm{x}) = \frac{1}{Z(\mathrm{x})} \prod_{\Phi_A \in G} \exp\left\{ \sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(\mathrm{y}_A, \mathrm{x}_A) \right\}$$

7.3 Suffix List:

a suffix list of Indian surnames inspire of using a specific surname. Similarly we can garner more and more features according to attributes of a language. Every feature function fi in CRF is having any real value on the basis of observation of the given language and these characteristics functions hold true for whole model distribution too.

Features of Telugu Language can be exploited for development of a good Named Entity Recognizer. Some features considered are as:

1. Some specific suffixes
2. Context Feature
3. Context Word List
4. Part of Speech of Words etc.

CRF has capability to introduce these features as binary value which makes it more useful for such problem.

7.3.1 Features:

1. Context Features: Preceding and following words of the current target word is very helpful for identifying NE category of the word. With identification of optimal window size of tokens we can get good results. For our experiment we have taken window size of five[2].
2. Context Pattern: Every language uses some specific patterns which may act as clue words and the list of this type of words is called as Context Lists. Such a list is compiled after analyzing Telugu text e.g. maMtri naayuDu,reDDi,adhyakshuDu etc for identification of person names and similarly for identification of places jilla, graamamu, bad, nagaraM etc.
3. Part-of-speech Features: Named Entities will fall in Noun Phrases and these boundaries can be found with help of Part of Speech category. Usually Verbs and Post-Positions denote the boundaries of such chunks. Some set of tags will give clue of being a word as NE.

External Resources (GAZ) : In order to measure the impact of using external resources in the NER task we have used A NERgazet which consists of three different gazetteers, all built manually using web resources:

(i) Location Gazetteer: this gazetteer consists of 22,000 names of villages, mandalas, Dicts, cities, in Andhra Pradesh found in the Telugu wikipedia and cities and states and countries found in other websites

(ii) Person Gazetteer: this was originally a list of 2000 complete names of people found in wikipedia and other websites. After splitting the names into first names and last names and omitting the repeated names, the list contains finally 3,450 names;

(iii) Organizations Gazetteer: the last gazetteer consists of a list of 400 names of companies,cricket teams,political party named and other organizations.

8. RESULT

We conducted experiments on a testing data of 150 sentences whereas model was created on 3000 sentences. Results on various combination's are tabulated in table 3, table 4, and table 5. Notations used in tables are as:

cw ->current word , pw ->previous word , nw->next word pw2->previous to previousword ,nw2 ->next to next word pt -> NE tag for previous word ,pt1-> NE tag for previous
to previous word cp –> Current pos tag , np -> next pos tag pp-> previouspos tag

Feature	Person	Location	Organization
pw , cw ,nw	57.8%	63.7%	40%
cw,pw,pw2,nw, nw2	60.4%	68.7%	48.7%
cw,pw,pw2,pw3 , nw, nw2, nw3	56.4%	67%	42.5%

TABLE3: Surrounding and current words combination:

Above table shows that window size of five gives optimum NE recognition. There would be no improvement in the accuracy even if the window size is increased further.

As shown in table4 Results are improved as compared to previous case because in this case we included NE tags which disambiguate some confusing classifications like in case of organization names

Feature	Person	Location	Organization
cw, pw, nw, pt	60.2%	64.6%	52.3%
cw, pw, pw2, nw, nw2, pt, pt2	62.2%	71%	52.00%
cw,pw, pw2,pw3, nw, nw2,nw3, pt, pt2,pt3	61.2%	69%	46%

TABLE 4: After adding NE tags:

Features	Person	Location	Organization
cw, pw, pp, nw, np, pt	66.7%	69.5%	58%
cw, pw, pp, pw2, pp2	66.3%	68%	58%

TABLE 5: After adding POS Tags:

Above table describes if we add pos tags in our word window, results could be improved further.

Above all results show that accuracy in case of organization, whatever combination we have taken ,is quite low compared to other NEs. It is because in most of cases organizations are multi word and even some cases comprise of Person Name, Location Name too. After inducting Part of Speech tag in feature, results get improved which justifies the approach chosen for embedding the features of language.

9. CONCLUSION

It is observed from our experiments and works done by other researchers too that CRF based system can be a viable solution if we identify and exploit features available in languages properly. It is also evident from experimental results (Table 5) that POS tags are playing an important role for getting better result. Our result is base on our training data and testing data. Most of the Indian languages gold standard data is not available because of unstable transliteration methods are used develop the training and testing data.

Future works includes increasing the relevant corpus size, induction of some more classification tags, and identification of some more features for Hindi Language. Inconsistency in the writing style e.g. telugu deesaM(Telugu Desam) and telugudeesaM (TeluguDesam) is the current limitation of the system, which can be handled too some extent by incorporating spelling normalization. Better classification of NEs can be achieved by induction of nested tag set. Rules can be crafted for identification and classification for the time, date, percentage, currency etc.

10. REFERENCES

[1] Asif Ekbal et. al*. “Language Independent Named Entity Recognition in Indian Languages”.* IJCNLP, 2008.

[2] Prasad Pingli et al. *“A Hybrid Approach for Named Entity Recognition in Indian Languages”.* IJCNLP, 2008.

[3] Lafferty, McCallum, et al. *“Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”.* 2001 .

[4] Himanshu Agrawal et. al*. “Part of Speech Tagging and Chunking with Conditional Random Fields”.* IJCNLP, 2008

[5]. Lafferty J., McCallum A., and Pereira F. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. In Proceedings of the Eighteenth International Conference on Machine Learning. 2001.

[6]. CRF++: Yet Another CRF toolkit http://crfpp.sourceforge.net/ (accessed on 13 rd Feb 2009)

[7] http://en.wikipedia.org/wiki/Named_entity (accessed on 11th Feb 2009)

[8]Navbharat Times http://navbharattimes.indiatimes.com (accessed on 11th Feb 2009)

[9] Chinchor, N. 1997. MUC-7 *Named entity task definition*. In Proceedings of the 7th Message Understanding Conference (MUC-7)

[10] Finkel, Jenny Rose, Grenager, Trond and Manning, Christopher. 2005. *“Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling.”* Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

[11] Kim, J. and Woodland, P.C. (2000a) *“Rule Based Named Entity Recognition”.* Technical Report CUED/F-INFENG/TR.385, Cambridge University Engineering Department, 2000.

[12] Malouf, Robert.2002 *Markov models for language-independent named entity recognition*. In Proceedings of CoNLL-2002 Taipei, Taiwan, pages 591-599.

[13] Pramod Kumar Gupta, Sunita Arora, *An Approach for Named Entity Recognition System for Hindi: An Experim-ental Study*, Proceedings of ASCNT – 2009, CDAC, Noida, India, pp. 103 – 108

[14] T. W. Anderson and S. Scolve, *Introduction to the Statistical Analysis of Data*. Houghton Mifflin, 1978.

[15] Kristjansson T., Culotta A., Viola P., and McCallum A. 2004. *Interactive Information Extraction with Constrained ConditionalRandom Fields.* In Proceedings of AAAI-2004.

[16] D. Roth and W. Yih. *Integer linear programming inference for conditional random fields*. In Proc. of the International Conference on Machine Learning (ICML), pages 737–744, 2005

[17] Zobel, Justin and Dart, Philip. 1996. *Phonetic string matching: Lessons from information retrieval.* In Proceedings of the Eighteenth ACM SIGIR International Conference on Research and Development in Information Retrieval, Zurich, Switzerland, August 1996, pp. 166-173.

[18]. Li W. and McCallum A. 2003. *Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction. In Special issue of ACM Transactions on Asian Language Information Processing*: Rapid Development of Language Capabilities: The Surprise Languages.

[19]. F. Sha and F. Pereira. *Shallow parsing with conditional random fields. roceedings of Human Language Technology,* NAACL 2003, 2003.

[20]. D. Pinto, A. McCallum, X. Wei, and W. B. Croft. *Table extraction using conditional random fields.* Proceedings of the ACM SIGIR, 2003.

[21]. Charles Sutton,Andrew McCallum, *An Introduction to Conditional Random Fields for Relational Learning*, Department of Computer Science University of Massachusetts, USA

[22]. Paul Viola and Mukund Narasimhan. *Learning to extract information from semistructured text using a discriminative context free grammar*. In Proceedings ofthe ACM SIGIR, 2005.

[23]. G.V.S.Raju, B.Srinivasu, S.V.Raju and Kumar*, Named Entity Recognition For Telugu using maximum entropy Model* , Journal of Theoretical and Applied Information Technology (JATIT), Vol-13, No-2, pages 125-130.

CALL FOR PAPERS

Journal: International Journal of Computational Linguistics (IJCL)
Volume:2 **Issue:**
ISSN:2180-1266
URL: http://www.cscjournals.org/csc/description.php?JCode=IJCL

Computational linguistics is an interdisciplinary field dealing with the statistical and/or rule-based modeling of natural language from a computational perspective. Today, computational language acquisition stands as one of the most fundamental, beguiling, and surprisingly open questions for computer science. With the aims to provide a scientific forum where computer scientists, experts in artificial intelligence, mathematicians, logicians, cognitive scientists, cognitive psychologists, psycholinguists, anthropologists and neuroscientists can present research studies, International Journal of Computational Linguistics (IJCL) publish papers that describe state of the art techniques, scientific research studies and results in computational linguistics in general but on theoretical linguistics, psycholinguistics, natural language processing, grammatical inference, machine learning and cognitive science computational models of linguistic theorizing: standard and enriched context free models, principles and parameters models, optimality theory and researchers working within the minimalist program, and other approaches. IJCL is a peer review journal and a bi-monthly journal.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCL.

IJCL List of Topics:
The realm of International Journal of Computational Linguistics (IJCL) extends, but not limited, to the following:

- Computational Linguistics
- Computational Theories
- Formal Linguistics-Theoretic and Grammar Induction
- Language Generation
- Linguistics Modeling Techniques
- Machine Translation
- Models that Address the Acquisition of Word-order
- Models that Employ Statistical/probabilistic Gramm
- Computational Models
- Corpus Linguistics
- Information Retrieval and Extraction
- Language Learning
- Linguistics Theories
- Models of Language Change and its Effect on Lingui
- Models that Combine Linguistics Parsing
- Models that Employ Techniques from machine learnin

- Natural Language Processing
- Speech Analysis/Synthesis
- Spoken Dialog Systems
- Quantitative Linguistics
- Speech Recognition/Understanding
- Web Information

Important Dates

Volume: 2 **Issue:** 1

Paper Submission: January 31, 2011

Author Notification: March 01, 2011

Issue Publication: March / April 2011

CALL FOR EDITORS/REVIEWERS

CSC Journals is in process of appointing Editorial Board Members for ***International Journal of Computational Linguistics (IJCL)***. CSC Journals would like to invite interested candidates to join **IJCL** network of professionals/researchers for the positions of Editor-in-Chief, Associate Editor-in-Chief, Editorial Board Members and Reviewers.

The invitation encourages interested professionals to contribute into CSC research network by joining as a part of editorial board members and reviewers for scientific peer-reviewed journals. All journals use an online, electronic submission process. The Editor is responsible for the timely and substantive output of the journal, including the solicitation of manuscripts, supervision of the peer review process and the final selection of articles for publication. Responsibilities also include implementing the journal's editorial policies, maintaining high professional standards for published content, ensuring the integrity of the journal, guiding manuscripts through the review process, overseeing revisions, and planning special issues along with the editorial team.

A complete list of journals can be found at http://www.cscjournals.org/csc/byjournal.php. Interested candidates may apply for the following positions through http://www.cscjournals.org/csc/login.php.

Please remember that it is through the effort of volunteers such as yourself that CSC Journals continues to grow and flourish. Your help with reviewing the issues written by prospective authors would be very much appreciated.

Feel free to contact us at coordinator@cscjournals.org if you have any queries.

Contact Information

Computer Science Journals Sdn BhD
M-3-19, Plaza Damas Sri Hartamas
50480, Kuala Lumpur MALAYSIA

Phone: +603 6207 1607
+603 2782 6991
Fax: +603 6207 1697

BRANCH OFFICE 1
Suite 5.04 Level 5, 365 Little Collins Street,
MELBOURNE 3000, Victoria, AUSTRALIA

Fax: +613 8677 1132

BRANCH OFFICE 2
Office no. 8, Saad Arcad, DHA Main Bulevard
Lahore, PAKISTAN

EMAIL SUPPORT
Head CSC Press: coordinator@cscjournals.org
CSC Press: cscpress@cscjournals.org
Info: info@cscjournals.org

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA