IJCSN International Journal of Computer Science and Network, Volume 4, Issue 5, October 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

Altered RELAN Algorithm to Display a Minimum of One Suffix and One Stem of Gujarati Language in Booting Screen at Booting Process in Linux based Mobile Operating System

[1] Milan Bhatt, [2] Prashant Dolia

[1] Mahadev Desai Grameva Mahavidyalay, Gujarat Vidyapith
Sadra, Guajrat 382320, India

[2] Department of Computer Science, Maharaja Krishnakumarsinhji Bhavnagar University
Bhavnagar, Gujarat 364002, India

Abstract - There is a no any display a Gujarati language character to booting screen in Linux based hand held devices, So Information technology to display a minimum of one suffix and one stem of Gujarati language in booting screen at booting process in Linux based mobile operating system. A immense cultural transition and transformation is taking place today in the way of Gujarati regional people, how to access, learn, and interact with information looking booting process in regional language to easy to understanding a star up information. Without doubt, has enormous power to improve how village people live and work using mobile.

Keywords - *Gujarati Language, Regional Language, Booting Screen, Suffix, Stem, RELAN, Linux Mobile.*

1. Introduction

At the starting of the configuration of a new mobile devices, mobile operating system's gives the message to the user in English language. So, ruler area people or specific domain regional language people can't understand English language and may not be understand the different messages properly at starting of mobile at first time which is given by the mobile operating system. Most all the people are working with the mobile technology in different way, domain peoples are uses the different mobile with specific operating system like Android, Windows, Asha, and Blackberry etc. But now a days almost booting process of operating systems in English Language. So new proposed booting process and its component and its process (different messages) gives in regional language.

2. Language Processing

We refer to the collection of language processor components engaged in analyzing a source program as the analysis phase of the language processor. Components engaged in synthesizing a target program constitute the synthesis phase.

Language Processing = Analysis of Source Programme + Synthesis of Target Programme.

Lexical analysis builds a descriptor, called a token, for each lexical unit. A token contain two fields—class code, and number in class, class code identifies the class to which a lexical unit belongs, number in class is the entry number of the lexical unit in the relevant table.
Syntax analysis processes the string of tokens built by lexical analysis to determine the statement class, e.g. assignment statement, if statement, etc.

Semantic analysis adds information a table or adds an action to the sequence. It then modifies the IC to enable further semantic analysis. The analysis ends when the tree has been completely processed.

3. Booting Process with Language Processor

Transferring the booting process of mobile devices English language to regional language booting process need some language processor for representation of an algorithm in a source language to and produces as output of target regional language. With the used of Assembler, Compilers, Pre-processor, Interpreters and Disassembler.

4. Proposed Structure of RELAN

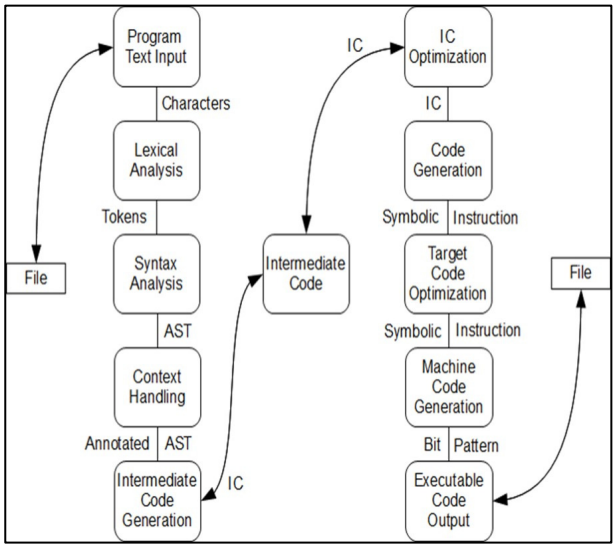

Fig. 1 Conceptual Structure of Language Processing

5. Morphology with Gujarati

Gujarati language has three genders (masculine, neuter rand feminine), two numbers (singular and plural) and three cases (nominative, oblique/vocative and locative) for nouns. The gender of a noun is determined either by its meaning or by its termination. The nouns get inflected on the basis of the word ending, number and case. The Gujarati adjectives are of two types – declinable and indeclinable. The declinable adjectives have the termination –ũ in neuter absolute.

The masculine absolute of these adjectives ends in -o (ો) and the feminine absolute in -ī (ી). For example, the adjective (mārũ - good) takes the form મારૂ (mārũ), મારો (māro) and મારી (mārī) when used for a neuter, masculine and feminine object respectively.

These adjectives agree with the noun they qualify in gender, number and case. The adjectives that do not end in -ũ in neuter absolute singular are classified as indeclinable and remain unaltered when affixed to a noun.

Table 1: Gujarati Moprhology

Gender	***Singular ()***	***Plural ()***
Masculine	(taar + o) or (tāro)	(taa + i) or (tārā)
Feminine	(taar + i) or (tārī)	(taar + i) or (tārī)
Neuter	(taar + uN) or (tārũ)	(taar + āN) or (tārā)

6. Altered RELAN

Step 1: Generate an object of obtain the optimal split position for each only two stem and only one suffix Gujarati word in the word list provided for training face data input stream and buffer_reader classes respectively "data_input_stream', 'buff_read'.

File_writer guj_char = newFile_writer("/usr/src/linux-source 2.6.8/kernel/guj_char.c");
{ stem1 + suffix1, stem2 + suffix2, stem3 + suffix3, ... , steml + suffixl }

guj_char -> guj_word [2] [N] array // separating character from text

STEP 2: Repeat Step 1 until the optimal split positions of all the words remain unchanged.

Loop { f(i) = i * log (freq(stemi)) + (L – i) * log (freq(suffixi)) } // i : Split position (Varies from 1 to L) and L : Length of the Word

STEP 3: Generate signatures using the stems and suffixes generated from the training phase.

1St Loop { // To get every character from string [f(i)]
2nd Loop { // To get suffix from the string [f(i)]
} // 2nd Loop
} // 1st Loop

STEP 4: Discard the signatures which contain either only one stem or only one suffix.

class buff_read closed
char stem; char suffix;
write to "guj_char.c ";

```
if ( guj_char==”stem”);
        write to “guj_char.c ”
else if ( guj_char==”suffix”);
        write to “guj_char.c ”
close();
```

During the RELAN phase, I try to obtain the optimal split position for each word present in the Gujarati word list provided for training. I obtain the optimal split for only one word and only two suffixes by taking all possible splits of the word and choosing the split which maximizes the function given in equation as the optimal split position. The suffix corresponding to the optimal split position is verified against the list of 59 Gujarati suffixes created by me. If it cannot be generated by agglutination of the hand crafted suffixes, then the length of the word is chosen as the optimal split position. i.e. the entire word is treated as a stem with no suffix.

{ stem1 + suffix1, stem2 + suffix2, stem3 + suffix3,……. stemL + suffixL }

મીનૂ = { મી + નૂ, મ + ી + ન + ૂ, મીનૂ + NULL }
ગતી = { ગ + તી, ગ + ત + ી, ગતી + NULL }

7. Applying RELAN

The bootloader is the first software program that runs when a computer starts. It is responsible for loading and transferring control to the Linux kernel. The grub.conf file is available in /boot/grub/grub.conf. Also the /boot/grub/grub.conf file can also be referenced via the symbolic link file named /etc/grub.conf. In this boot loader the default font is unicode.pf2. Now set the Gujarati fonts in the boot loader your first check the Unicode of Gujarati fonts. Gujarati Lohit font’s Unicode number is set within 0A80 – 0AFF. There are 128 different character and suffixes.

Fig. 2 Gujarati Unicode

In this step you can create a new system calls in under kernel.h file and this file is available in /usr/src/linuxsource—2.6.8/kernel/<file_name>.c

Now generate the simple Gujarati character with the kernel file and linkage with the <file_name>.c with kernel.h

<file_name>.c = guj_char.c

```
#include <linux/stdio.h>
#include <linux/linkage.h>
#include <linux/kernel.h>
#include <asm/uaccess.h>
#include <asm/locale.h>
#define MAX_BUF_SIZE 4

asmlinkage int sys_guj_char(char _ _ usr *buff, int len){
char tmp[MAX_BUF_SIZE]; // tmp buffer to copy user's string into
int guj_stem_len;  // find how many stems in a string
int guj_suffix_len; // find how many suffixes in a string
```

a:

```
  if ( guj_char==”stem”); then
        write to “guj_char.c ”
  elif ( guj_char==”suffix”); then
        write to “guj_char.c ”


    close(); fi

File_writer guj_char = new File_writer(“/usr/src/linux-source—2.6.8/kernel/guj_char.c”;
Buffer_writer guj_char = new buffer_writer(guj_char);
class buff_read  closed
// Now Apply RELAN algorithm
char suffix;
char stem;
guj_char = { stem1 + suffix1, stem2 + suffix2, stem3 + suffix3, ..... , steml + suffixl } ;
guj_char -> guj_word [2] [N] array;
guj_char = i * log (freq(stemi)) + (L – i) * log (freq(suffixi));
printk(KERN_EMERG “Entering guj_char(). The len is %d\n”, len);
char guj_char_list;

if (len <= 2 || (len > MAX_BUF_SIZE)); then
printk((KERN_EMERG “Entering guj_char() failed: illegal len (%d) !”, len);
        return (-1);
return a; fi
// copy buff from user space into a kernel buffer
if (copy_from_user(tmp, buff, len); then
printk(KERN_EMERG “Entering guj_char() fail: copy_from_user() error”);
        return (-1);
return a; fi
tmp[len] = '\0';
printk( KERN_EMERG “ guj_char() from %s. \n”, tmp);

if (!setlocale(LC_CTYPE, "")) {

    fprintf(stderr, "Can't set the specified locale! " "Check

LANG, LC_CTYPE, LC_ALL.\n");

    return 1 }

  printf("%ls\n", L "Bhatru – ascii(0AC3) ");
  return 0;
return (0);
close()
}
```

Now this guj_char file linkage with the modified grub.conf file as name is modgrub.conf.

The changes will come into effect right after we run 'sudo grub-mkconfig -o /boot/grub/grub.cfg' to write the new modgrub.cfg file.

```
# Manually re-load gfxterm
# update-grub generates a grub.cfg that thinks grub can see the SD card reader
# (where "/" lives) at boot time to get the font for gfxterm from /usr/share/...
# So we do manually here what update-grub fails to do automatically in grub.cfg
insmod part_msdos
insmod ntfs
set root='(hd0,msdos2)'
if loadfont /linux_boot/grub/fonts/Lohit-Gujarati.ttf ; then
set locale_dir=($root)/linux_boot/grub/locale
set lang=en_US
set gfxmode=1600x768x16
set gfxterm_font=Lohit-Gujarati.ttf
load_video
insmod gfxterm
insmod gettext
insmod png
terminal_output gfxterm
background_image /linux_boot/grub/MIDO.png
fi
```

Now grub2 copy Lohit-Gujarati to /boot/grub/fonts.
As long as that's there, then all you need to do is one time run this:

```
grub-editenv - set feature_default_font_path=y
```

Now update-grub still produces a grub.cfg that points to /usr/share , but now this part gets invoked instead of the part that tries to use the unreachable path:

```
if [ x$feature_default_font_path = xy ] ; then
  font=Lohit-Gujarati.ttf
else
```

There does seem to be an undocumented variable GRUB_FONT that you can put in /etc/default/grub

```
root@mido:~# update-grub
```

Now reboot the computer and check the maximum two Gujarati characters and maximum two suffixes are display in the booting process.

So during my this research paper, I try to solve Specific domain people can understand at the start up (booting process) the Nokia N900 Linux mobile in regional language. Using this entire work based on Linux operating system and MAEMO operating system which is based on Debian series as well as kernel programming. During this work I have made RELAN Algorithm and apply to the Nokia N900 mobile device and work properly and mobile phone was hang up. Then required to modified RELAN algorithm and some changes in guj_char.c file.

8. Display a Gujarati Character

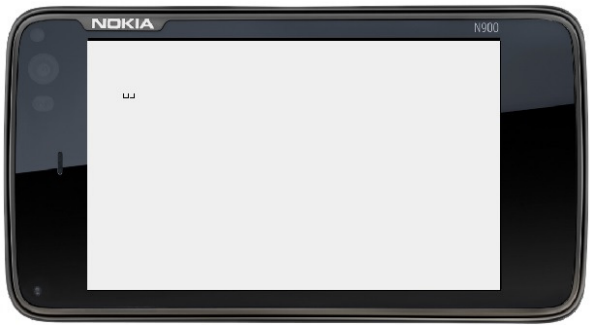

Fig. 3 Gujarati Character

9. Conclusion

A immense cultural transition and transformation is taking place today in the way of Gujarati regional people, how to understand, access, learn, and interact with information looking booting process in Gujarati language for the Linux based mobile Operating System.

References

[1] Milan S., Prashant M., "A Proposed to Display any One Character of Regional Language for Booting Screen of Linux Based Mobile Operating System Using RELAN Algorithm", International Journal of Computer Science and Technology [IJCST], IJCST Vol. 6, Issue 2, April - June 2015, pp: 314-317

[2] Milan S., Prashant M., "A Proposed RELAN Algorithm for Booting Process in Regional Language for Linux Based Mobile Operating System", INDIAN JOURNAL OF APPLIED RESEARCH, Vol. 5, No. 2, 2015, pp: 203-207

[3] Milan S., Prashant M., "Proposed Booting screen and Architecture in regional language for Linux based mobile devices", International Journal of Computer Applications, NCETICT 2013, pp.: 28-33

[4] Milan S., Prashant M., "Linux સંચાલિત Mobile માટે ગુજરાતી ભાષાનું સૂચિત સ્થાપત્ય (Architecture) તથા ગુજરાતી ભાષામાં સૂચિત Desktop અને વિવિધ Software."Shabd Braham-International Research Journal of Indian Languages, Vol. 1, No.9, pp.: 67-73

[5] Jin Zhou, Brian Demsky, "Memory Management for Many-Core Processors with Software Configurable Locality Policies", International Symposium on Memory Management, 2012.

[6] Vivek D., Vijay W., Helonde J., "Characterizing the Memory Management for Improving the Performance of Embedded system used in Wireless Sensor Networks ", IJCA Proceedings on International Conference in Computational Intelligence, 2012.

[7] Raghavan S., Mooney, R. J., Ku. H, "Learning to read between the lines using Bayesian Logic Programs", ACL 2012.

[8] Savvas G., Nicholas B., "Joint Transmitter Power Control and Mobile Cache Management in Wireless Computing", IEEE Transactions on Mobile Computing, Vol. 7, No. 4.

[9] Prasenjit M., Madar M., Swapan P., Gobinda K., Pabitra M., Kalyankumar D., "YASS: Yet another suffix stripper", ACM Transactions on Information Systems, Vol. 25, No. 4, pp 18-38, 2007.

[10] Dagan I., Greental I., and Shnarch E., "Semantic inference at the lexical-syntactic level Bar-Haim", 22nd AAAI Conference on Artificial Intelligence, 2007.

[11] Creutz, Mathis, Krista L., "Unsupervised models for morpheme segmentation and morphology learning. Association for Computing Machinery Transactions on Speech and Language Processing", 2007.

[12] John G., "An algorithm for unsupervised learning of morphology", Natural Language Engineering, Vol. 12, No. 4, pp 353-371, 2006.

[13] Amaresh P., Tanveer S., "An unsupervised Hindi stemmer with heuristic improvements", 2nd Workshop on Analytics for Noisy Unstructured Text Data, pp 99-105.

Author Profile

Milan S. Bhatt received his B.Sc. degree in Industrial Chemistry from Bhavnagar University, Bhavnagar, India, in 2002, the M.C.A degree from Bhavnagar University, Bhavnagar, India, in 2005, and the currently pursuing Ph.D. degree from Gujarat Technological University, Ahmedabad. He was a teaching assistant, assistant professor, in MCA and M.Sc. IT Department with SSCCS, Bhavnagar, in 2006 to 2011 respectively. He was also teaching assistant in BCA with IGNOU. His research interests include Kernel Programming, Linux Operating System and Network with Windows and Linux Operating System. Also published 6 books of related to computer science and also published 6 research paper in national and international peer review journals. At present, he is engaged Programmer at MD Gramseva Mahavidyalay, Sadra –Gujarat Vidyapith, Ahmedabad.

Prashant M. Dolia received his B.Sc. degree in Physics from Bhavnagar University, Bhavnagar, India, the M.C.A degree from Bhavnagar University, Bhavnagar, India, and the complete the Ph.D. He was Project Officer, in MCA Department, Bhavnagar University,

Bhavnagar, in 1998 to 2000 respectively. He was lecturer in MCA Department, Bhavnagar University, Bhavnagar. His research interests include Web Intelligence, Data Mining & Warehousing, Wireless Technology, Linux Platform & Kernel Programming. Also published 9 books of related to computer science and also published 21 research paper in national and international peer review journals. At present, he is engaged Associate Professor at Department of Computer Science, Maharaja Krishnakumarsinhji Bhavnagar University, Bhavnagar.