Information Retrieval in Indian Languages: A Case Study of Plural Resolution in Telugu Language

Dimple Patel [1] and Devika P. Madalli [2]

[1] Mahatma Gandhi National Institute of Research and Social
Action, Hyderabad, India
dimple@drtc.isibang.ac.in

[2] Documentation Research & Training Centre, Indian Statistical
Institute, Bangalore, India
devika@drtc.isibang.ac.in

Abstract. The paper deals with developing stemming algorithms that could be applied to develop a search and retrieval system for digital documents in Telugu - a language belonging to the South-central branch of the Dravidian languages. A set of stemming algorithms is developed to index and search for documents in the Telugu language. The algorithms developed demonstrate successfully that stemming algorithms for Indian languages can be developed for increasing the search efficiency in Indian languages.

1 Introduction

English language documents dominate the Web when compared to other languages of the world (Wikipedia, 2006). It is a kind of paradox that English stands third with Chinese and Hindi at the first and second place among the world's spoken languages. It is only recently that, there is evidence of ever-growing literature in Indian languages on the Web. One of the offshoots of Internet is the development of Digital Libraries. Many libraries and institutions across India have realized the importance and role of Digital Libraries in disseminating information to their clientele. If one studies the developmental activities in Digital Libraries in India, one observes the fact that in India many institutions deal with documents in Indian languages apart from English language documents. This is especially true of Universities, where the presence of few Indian language departments is not uncommon.

Most of the search engines can index and search English documents and some European languages like Altavista and Google support Greek, French, German, etc. Many search engines and digital library software like DSpace do support Indian scripts. However, they do not support stemming algorithms for Indian languages, consequently, one can only make exact keyword search,

ARD Prasad & Devika P. Madalli (Eds.): ICSD-2007, pp. 397-404, 2007

like Google has come up with search pages in five Indian languages i.e. Hindi, Bengali, Telugu, Marathi and Tamil. The complexities of grammar, syntax, and morphology and script of Indian languages are the main barriers in developing search algorithms for these languages. The approaches and methodology adopted for English language are not adequate for processing Indian language queries.

2 Characteristics of Indian Languages

India is a multi-lingual country with twenty two constitutionally recognized languages. However, in spite of their diversities, all most all the scripts are derived from *Brahmi* and the order of alphabets in all the scripts is similar. They also share some common characteristics like, common phonetic based alphabet; non-linear and complex scripts; word order free; there are no cases (upper or lower) in Indian scripts. A very peculiar feature of Indian languages is that though vowels can occur independently at the beginning, they do not occur independently within a word or as the last character of a word.

3 Encoding Standards for Indian Languages

The two main standards in character representation of Indian languages are ISCII and Unicode.

3.1 Indian Standard Code for Information Interchange (ISCII)

Indian Script Code for Information Interchange (ISCII) is an 8-bit code. It covers 10 Indic scripts (Devanagari, Gujarati, Punjabi, Bengali, Assamese, Oriya, Telugu, Tamil, Malayalam, Kannada). ISCII uses extended ASCII and uses last 128 characters position for characters representation in Indic scripts. The arrangement of characters is phonetic. (Appendix 1)

3.2 Unicode

The Unicode Consortium was initiated in January 1991, under the name Unicode, Inc., to promote the Unicode Standard as an international encoding system for information interchange, to aid in its implementation, and to maintain quality control over future revisions. (Addison Wesley, 2004) (Appendix 2) Currently, Unicode is in version 4.1.0. The Unicode standard provides with three encoding formats: UTF-8, UTF-16 and UTF-32. Any one of these forms can be used to represent the Unicode characters. Each of these is used in different environments. The default encoding form of Unicode is UTF-16. Operating System level support for Unicode encoding of Indian language scripts is available both on Windows XP and Linux. Unicode fonts for many of the Indian languages are now available. In addition, HTML supports Unicode.

4 Study of Telugu language

Telugu is one of the twenty two officially recognized languages of India. Telugu is a member of the Telugu languages which are part of the South-central branch of the Dravidian languages (the other Telugu languages being *Chenchu, Savara* and *Waddar*).

In Telugu language the stem/root of a word is known as *'Dhaathu'*. The *Dhaathu* or the stem undergoes many modifications in cases of plural/singular forms, gender, tense, dative and accusative cases, animate and inanimate objects. This is explained with an example below. The example discusses the postpositions i.e. Dative కి / కు (*ki/ku*) and Accusative ని / ను (*ni/nu)* suffixes.

The Dative suffixes *ki* and *ku* denote *'to'* or *'for'* to the basic stems of words. The Accusative suffixes *ni* and *nu* denote the object of the sentence. When the object is an inanimate object (like *illu*, meaning house, in the example), the Accusative case is same as the nominative. Its use in case of inanimate objects is optional. But, nouns denoting animate objects (like *snehithudu*, meaning friend in the example) have to take Accusitive suffix. (*see* Table 1).

Singular		
Basic stem (nominative)	ఇల్లు illu (house)	స్నేహితుడు snehithudu (friend)
Oblique stem (genitive)	ఇంటి inti (of a house)	స్నేహితుడి snehithudi (of a friend)
Accusative	ఇల్లు illu (house)	స్నేహితుణ్ణి snehithunni [or] స్నేహితుడిని snehithudini (friend)
Dative	ఇంటికి intiki (to a house)	స్నేహితుడికి snehithudiki (to a friend)
Plural		
Basic stem (nominative)	ఇళ్ళు iLLu (houses)	స్నేహితులు snehithulu (friends)
Oblique stem (genitive)	ఇళ్ళ iLLa (of houses)	స్నేహితుల snehithula (of friends)
Accusative	ఇళ్ళు iLLu (houses)	స్నేహితులని /ను snehithulani/nu (friends)
Dative	ఇళ్ళకి / కు iLLaki/ku (to houses)	స్నేహితులకి / కు snehithulaki/ku (to friends)

Table 1. Example of stem/root word modifications in Telugu

Many variations and transformations occur in a word in Telugu due to *sandhi* formations, *vibhakthis* and *samasas*. All these variations and transformations have to be analyzed by morphological analysis of the word to arrive at the Basic stem of the word. A comparative study of the search algorithms in English and Telugu is presented in the table below (*see* Table 2):

Search algorithm	**English**	**Telugu**
Representation	ASCII, Unicode compatible	ISCII and Unicode compatible
Exact search	Possible	Possible
Truncation	Simple	Requires morphological analysis
Spelling variations	British and American Eg. Colour & color	No spelling variants.
Variant words	Already identified Eg. Manage, managed, managing, management,	Requires to be identified. Eg. Ramunichetha, ramunivalla
Thesaurus	Readily available, general as well as subject-specific	Need to be explored.
Embedded words	Morphological analysis of prefixes, suffixes and roots.	More complicated because of *'vibhakthis'*, *'samasas'*, *'sandhis'*.
Tolerance to error	Books on common spelling mistakes available readily	Not readily available.
Transliteration	Complex	Fairly easy within Indian Languages, though not without problems.

Table 2. Comparative study of search algorithms in English and Telugu

5 Stemming Algorithm for Telugu

A stemming algorithm is a process of linguistic normalization, in which the variant forms of a word are reduced to a common form, called the *root stem.* This work deals with the problem of plural resolutions in Telugu language. A set of rules has been adapted to develop algorithms for plural resolution in Telugu language. The corpus database used as testbed is in UTF-8 encoding format. The algorithms developed demonstrate successfully that stemming algorithms for Indian languages can be developed for increasing the search efficiency in Indian languages.

5.1 The Approach

The simplest algorithm for plural formation in Telugu is to add the suffix *–lu* to each word. But, this does not work in all cases. It fails in many special cases and irregular plural nouns. For example, చెట్టు *cheTTu* 'eye' → చెట్లు

cheTlu ‘eyes’ and not చెట్టులు *cheTTulu*. Complex algorithms dealing with specific suffixes can be developed, but still there will be exceptions. For example, words ending in short vowel –*i* change to –*u* followed by the plural –*lu*. For example, *baawi* ‘well’ → బావులు *baawulu*. But this rule changes when words ending with –*i* and *i’s* occuring in non-initial open syllables become *u’s* when followed by the plural –*lu*. For example, మనిషి *maniSi* ‘man’ → మనుషులు *manuSulu* ‘men’.

Therefore, the algorithm for plural formation can be categorized into three kinds:

1. Universal Default
2. Rule-based Suffix formation
3. Specific exceptional cases

Approach 1: Universal Default

Here the general rule for plural formation is applied. The most commonly and frequently occurring plural suffix in Telugu is –*lu*. But as this rule does not take care of special cases of nouns, it is dealt in the last. This rule is applied only in the cases where the other specific rules are inapplicable.

Approach 2: Rule-based Suffix formation

There will be many exceptions to the default rule discussed above. However most of these exceptions are still regular i.e. their pattern is predictable, but are specific to a particular word suffix. For example, a geminate (double) consonant becomes single before another consonant across a morph boundary.

గుడ్డు [guDDu] ‘egg’ → గుడ్లు [guDLu] ‘eggs’
చెట్టు [ceTTu] ‘tree’ → చెట్లు [ceTLu] ‘trees’

Approach 3: Specific exceptional Cases

The third approach is to deal with specific cases which are exceptional to the above two approaches. For instance, the classification of nouns based on their different phonological behaviour in plural formation as suggested by Krishnamurti and Gwynn (Krishnamurti & Gwynn, 1985). For example, కన్ను *kannu* ‘eye’ → కండ్లు *kaNDLu [or]* కళ్ళు *kaLLu*, but పన్ను *pannu* ‘tax’ → పన్నులు *pannulu*. There is no way to distinguish these two types of stems except by assigning them to two different stem classes.

Hence, the approach taken in this work to develop algorithms was to first take care of the word stems belonging to Classes I to VI. Once, these definite number of stems were taken care of the general rules as described below were applied to the rest of the words belonging to the unmarked class Class 0.

6 Conclusion

Due to the fact that grammar came much later than spoken language, exceptions in grammatical rules are natural. That is what has been observed in our study. Though, there are some set rules for plural formation in Telugu, there is also a large number of exceptions. For example, the same word can have more than one plural forms e.g. కండ్లు *kaNDlu* and కళ్ళు *kaLLu* are the alternate forms of the same singular noun కన్ను *kannu*, or, the same singular word having different meaning depending on context will form different plurals. For e.g. for the singular noun పన్ను *pannu* (which has two different meanings i.e. tooth as well as tax) forms the plural పళ్ళు *paLLu* (teeth) పన్నులు *pannulu* for the latter (i.e. tax).

Humans infer the semantics of a sentence even if the speaker does not pronounce the words distinctly. Indeed machines are not blessed with such intuitive learning. To make meaningful retrieval in Indian Languages search engines will have to understand the intricacies and nuances of the language. Though, this may not mean pragmatic language understanding as aimed in Natural Language Processing (NLP), but at least morphological understanding is essential. This work is a step towards this purpose. Of course translation of thoughts to action is a favorite indulgence that may come true one day language no bar!

References

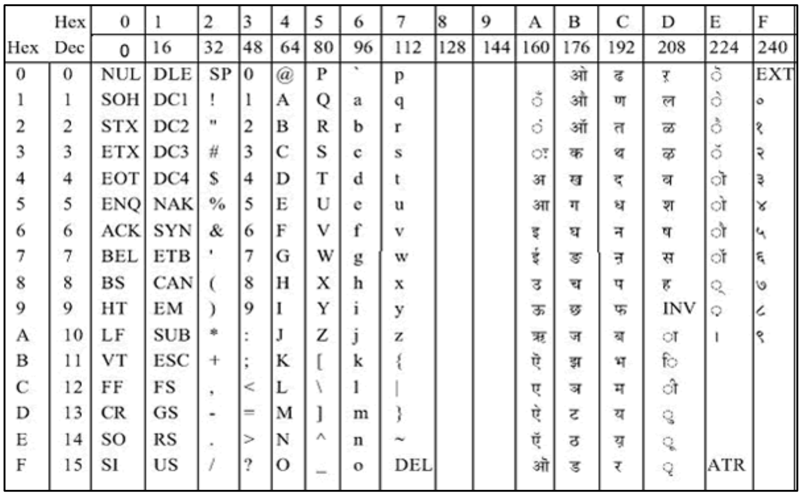
[1] Wikipedia. (2006). *English on the Internet*.
http://www.wikipedia.org
Accessed on 1st December 2006.

[2] Bureau of Indian Standards. (1991). *Indian Script Code for Information Interchange (ISCII), ISCII-91 or IS13194:1991.*

[3] Addison Wesley. (2004). *Unicode standard version 4.1.*
http://www.unicode.org/standard/standard.html
Accessed on 1st December 2006.

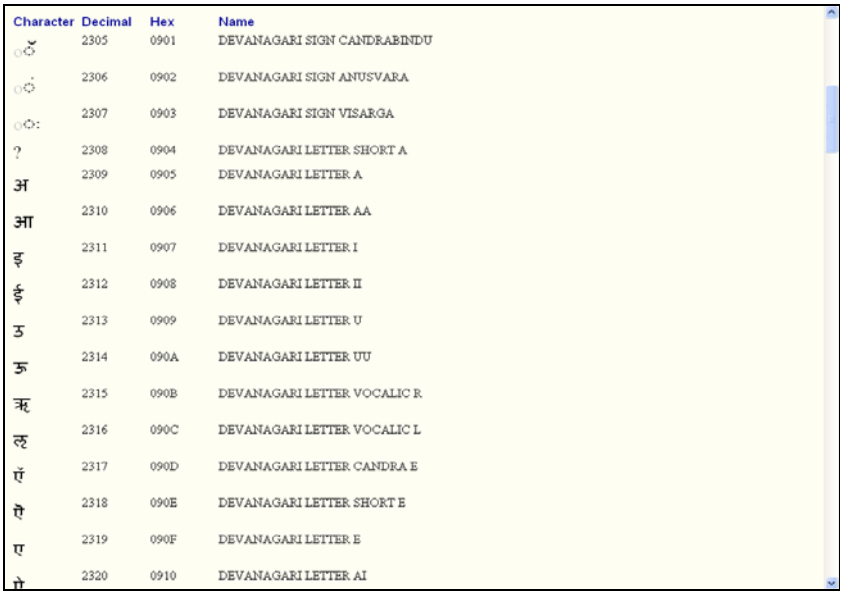
[4] Krishnamurti, Bhadriraju & Gwynn, J.P.L. (1985). *A Grammar of Modern Telugu*. Delhi: Oxford University Press.

Appendix 1
ISCII Table



	Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hex	Dec	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
0	0	NUL	DLE	SP	0	@	P	`	p				ओ	ढ	ऱ	◌ॆ	EXT
1	1	SOH	DC1	!	1	A	Q	a	q			◌ँ	औ	ण	ल	◌े	०
2	2	STX	DC2	"	2	B	R	b	r			◌ं	ऑ	त	ळ	◌ै	१
3	3	ETX	DC3	#	3	C	S	c	s			◌ः	क	थ	ऴ	◌ॅ	२
4	4	EOT	DC4	$	4	D	T	d	t			अ	ख	द	व	◌ॊ	३
5	5	ENQ	NAK	%	5	E	U	e	u			आ	ग	ध	श	◌ो	४
6	6	ACK	SYN	&	6	F	V	f	v			इ	घ	न	ष	◌ौ	५
7	7	BEL	ETB	'	7	G	W	g	w			ई	ङ	ऩ	स	◌ॉ	६
8	8	BS	CAN	(	8	H	X	h	x			उ	च	प	ह	◌्	७
9	9	HT	EM	)	9	I	Y	i	y			ऊ	छ	फ	INV	◌़	८
A	10	LF	SUB	*	:	J	Z	j	z			ऋ	ज	ब	◌ा	।	९
B	11	VT	ESC	+	;	K	[	k	{			ऎ	झ	भ	◌ि		
C	12	FF	FS	,	<	L	\	l	|			ए	ञ	म	◌ी		
D	13	CR	GS	-	=	M	]	m	}			ऐ	ट	य	◌ु		
E	14	SO	RS	.	>	N	^	n	~			ऍ	ठ	य़	◌ू		
F	15	SI	US	/	?	O	_	o	DEL			ऒ	ड	र	◌ृ	ATR	

Appendix 2
Unicode Code Chart for Devanagari Script



Character	Decimal	Hex	Name
◌ँ	2305	0901	DEVANAGARI SIGN CANDRABINDU
◌ं	2306	0902	DEVANAGARI SIGN ANUSVARA
◌ः	2307	0903	DEVANAGARI SIGN VISARGA
?	2308	0904	DEVANAGARI LETTER SHORT A
अ	2309	0905	DEVANAGARI LETTER A
आ	2310	0906	DEVANAGARI LETTER AA
इ	2311	0907	DEVANAGARI LETTER I
ई	2312	0908	DEVANAGARI LETTER II
उ	2313	0909	DEVANAGARI LETTER U
ऊ	2314	090A	DEVANAGARI LETTER UU
ऋ	2315	090B	DEVANAGARI LETTER VOCALIC R
ऌ	2316	090C	DEVANAGARI LETTER VOCALIC L
ऍ	2317	090D	DEVANAGARI LETTER CANDRA E
ऎ	2318	090E	DEVANAGARI LETTER SHORT E
ए	2319	090F	DEVANAGARI LETTER E
ऐ	2320	0910	DEVANAGARI LETTER AI