TELUGU SPELL-CHECKER

Uma Maheshwar Rao[1], G., Amba P. Kulkarni[2], Christopher[3] Mala and Parameshwari[4], K.

Center for Applied Linguistics and Translation Studies
University of Hyderabad
Hyderabad, India
[1]guraohyd@yahoo.com, {[2]ambapradeep, [3]efthachris, [4]cuteparamesh}@gmail

Abstract:

Spell Checker is an application which handles spelling errors and Spelling Variations (SV). All the misspelt words are marked and allowed for correction. This system also can be used as an editor where the text is checked for spelling errors and suggestion for correction are provided. Telugu is an agglutinating language and has a very complex morphology which is coupled with prolific sandhi or morphophonemics. The sandhi that is noticed in Telugu is not limited to internal but also external. Both consonantal and vocalic sandhi are common and well studied in Telugu [Krishnamurti, 1957, 1985]. To identify the specific sandhi type and split it appropriately is a very challenging task. External sandhi is a linguistic phenomenon which refers to a set of changes that occur at word boundaries. These changes are similar to phonological processes such as substition (modification by various means) deletion, and insertion. External sandhi is often orthographically reflected in Telugu. External sandhi in such cases, causes the formation of such forms which are morphologically unanalyzable, thus posing a problem for all kinds of NLP applications. In this paper, we discuss in detail the processes external sandhi in Telugu and the Computational tool the Spell Checker.

Key Words: Sandhi Splitter, Sandhi, Spelling Variation, Dashboard, Morphophonemics, Morphological Analyzer.

Introduction:

In editing or Text processing one of the most common applications is a *Spell-Checker.* A spell-checker is an application program that flags words in a document that are not spelled correctly and facilitates corrections. Words can be defined from morphophonemic, morphological, lexical, and orthographic perspective. Spell-checkers as stand-alone applications are capable of operating on a block of text, or as part of a larger application, such as a word processor, email client, or a search engine. Spell-checkers are the basic tools needed for word processing and document preparation. Designing a spell checker for Indian languages such as Telugu poses many new challenges not found in English, which complicates the design of the spell checker. The Telugu language is far different from the European languages in terms of their morphophonemics and word formation rules. Telugu is a Dravidian language with agglutinative morphology (Krishnamurthi & Gwynn, 1985). It must be noted that agglutination in its original formulation (Sapir, 1921), refers to the property of a one-to-one mapping between morphs/ morphemes and their functions. In Telugu, inflectional elements (which include different kinds of auxiliary verbs, postpositions, particles and case-markers) are always bound to the stem resulting in highly synthetic word forms. The number of possible verb forms for a verb stem in Telugu therefore, is very high running into millions, aggravating the task of the morph analyzer.

ex. pagalagoVttiMcipeVttamananivvaxalacukolekapowunnAnu.

Pagulu+a-koVttu+iMcu+i-peVttu+a-manu+a-ivvu+a-xaluvu+u-koVnu+a-le +aka-po+wunn+1,sg,any
be broken+a-strike+pt_ppl-keep+inf <--cause+inf-benefactive+inf-tell +inf-permit +inf-Think+c_ppl-reflexive+inf-neg-verbs+neg-verbs+neg.infl-go +pr.tense +1, sg

Even full morphological words can fuse together in Telugu resulting in complex forms which become morphologically unanalyzable.

ex. iMtikoVccAdu <= iMtiki +vaccAdu
UrikeVlYlAdu <= Uriki +veVlYlAdu

As this random concatenation results in longer strings they are again broken to be realized as smaller strings in the text inappropriate.

ex. iMti koVccAdu
and Uri keVlYlAdu

Such complexities in the morphology of Telugu point towards the need for a more exact approach while typifying them, similar perhaps, to the one espoused in Greenberg (1960). The existing algorithms and techniques that are being used to check the spelling and to generate efficient suggestions for misspelt words as in conventional spell-checker designs are not actually suitable for Telugu(Cf. Chavala). The Telugu language spell-checker rather needs a different algorithm and technique to achieve appropriate results. A rule-based approach for spell-checkers is preferred due to its morphological richness which usually involves a variety of phenomena such as morphophonemic variants, dialectal variants, classical and Modern dialectal variants, non-standard variants and misspelt forms (!unattested variants). In this respect a spell checker is theoretically interpreted for the first time as an effort in the creation of an acceptable standard text bringing it into invariance, in other words a procedure to move from variance to invariance.

This paper presents the novel design and implementation of a Telugu spell- checker. Morphological validation by a Morphological Analyzer is the core component of the Telugu Spell-Checking. Besides discussing complexities involved in spell checking of documents in Telugu, issues involving both orthography and morphology are discussed. A spell-checker designed on these lines has been developed. The architecture of the spell-checker and the spell-checking algorithm based on Morphological Analysis and Sandhi Splitter rules are outlined. It also includes lists of spelling variants obtained from spatio-temporal dialects of Telugu. A spell checker customarily consists of two parts: a set of routines for scanning the text (Morphological Analyzer and sandhi splitting rules) and identifying valid words, and an algorithm for comparing the unrecognized words and word parts against a known list of variantly spelled words and word parts.

Sandhi:

Briefly stated, sandhi refers to a set of morpho-phonological processes that occur at either morpheme or word boundaries. Two types of sandhi are identified in a language, viz. ***internal sandhi*** and ***external sandhi***.

Internal sandhi refers to word-internal morphonological changes that take place at morpheme boundaries during the process of word-formation.

An example of internal sandhi in English would be the positional variation of the negative morpheme ‘*in-*’ to give the allomorph ‘*im-*’ when it is prefixed to words that begin with bilabial sounds as in ’impossible’. Such processes lie obviously, within the domain of morphology. ***External sandhi***, on the other hand, refers to processes that apply word-externally i.e. across word boundaries. Examples of external sandhi formation in English are the well- known cases of *wanna /hafta /gotta* contractions where the verb combines with the infinitival ’to’ following it to give the contracted form. Note that external sandhi as seen in these examples need not always be reflected orthographically in English (’*want to*’ while writing, but spoken as ’*wanna*’).

Cases of external sandhi formation, have attracted special attention in generative phonology. A phonological phrase, therefore, is the domain within which external sandhi rules operate(Selkirk, 1981). In the dravidian languages, sandhi (both internal and external) have a wide-spread occurrence and are also orthographically reflected most of the time. External sandhi formation in Telugu leads to fusion of morphological words resulting in morphologically complex/unanalyzable forms. This poses a problem for all natural language processing applications such as POS-tagging, chunking, parsing, etc. that deal with written text. The task of tokenization becomes complex in these languages as tokens obtained through sentence splitting can contain more than one morphological word within them. Since external sandhi is a consequence of (orthographically visible) phonological processes occurring at the prosodic level, splitting such instances of sandhi can not fall within the purview of the morph analyzer. The task of splitting sandhi forms requires segmentation at a different level and should be treated as being distinct from morphological segmentation. Without this distinction between sandhi formation and other kinds of morphological changes, the task of morphological analysis in languages like Telugu becomes extremely complex (Uma Maheshwar Rao, 2002).

Data Organization:

To build a spell checker we need to build a sandhi splitter first. The linguistic data that is required to build this sandhi splitter is as follows.

1. Rule Format
2. Splitting Rules
3. Spelling Variation Rules
4. Proper Names

			Left		Right		Mocall		CONDITION	
S.No	Pattern	br_pt	Delete	Add	Delete	Add	Left	Right	Left	Right
1	Ak[uUAeo]	0	0	u	0	0	1	1	n,ti	n,0

2	Ak[uUAeo]	0	0	i	0	0	1	1	n,ti	n,0
3	Ak[uUAeo]	0	0	a	0	0	1	1	n,ti	n,0
4	Ak[uUAeo]	0	0	eV	0	0	1	1	n,ti	n,0
5	Aspax[au]	0	0	a	0	0	1	1	n,ti	n,0
6	Awmak	0	0	a	0	0	1	1	n,0	adj,0
7	vAx[iIueoa]	0	0	0	0	0	1	1	n,0	
8	vAx[iueAo]	0	0	i	0	0	1	1	n,0	
9	vAx[iueAo]	0	0	u	0	0	1	1	n,0	

Table-1 Rules format and the splitting rules

Currently there are about 250 split rules handcrafted and placed in the rule format followed by the break point, then delete or add segments on the left part of the pattern after the split and delete or add segments in the right part of the string resulting from split and call the morph for analyzing the left and the right strings and the specification of categories of these conditions if necessary.

Spell Variation Rules:

A series of mechanically collected, manually checked spelling variats manually checked spelling variants in the extracted and the corresponding collect equivalents one provided.

Ex: warvAwa,	waravAwa
warvAwa,	waruvAwa
Adapaducu,	Adabaducu
kriMxa,	kiMxa

Once a variant is identified by matching against the list an appropriate equivalent is suggested and then passed over to the morph for analysis.

Proper Nouns:

Usually texts abound in proper names ending in _Uru, _nagar, _reddi, _rAm, _pAdu, _puraM, _AlayaM etc. Such words are routed through NER correctly identifies these and then passed on to morph for anlysis.

Implementation of Spell Checker:

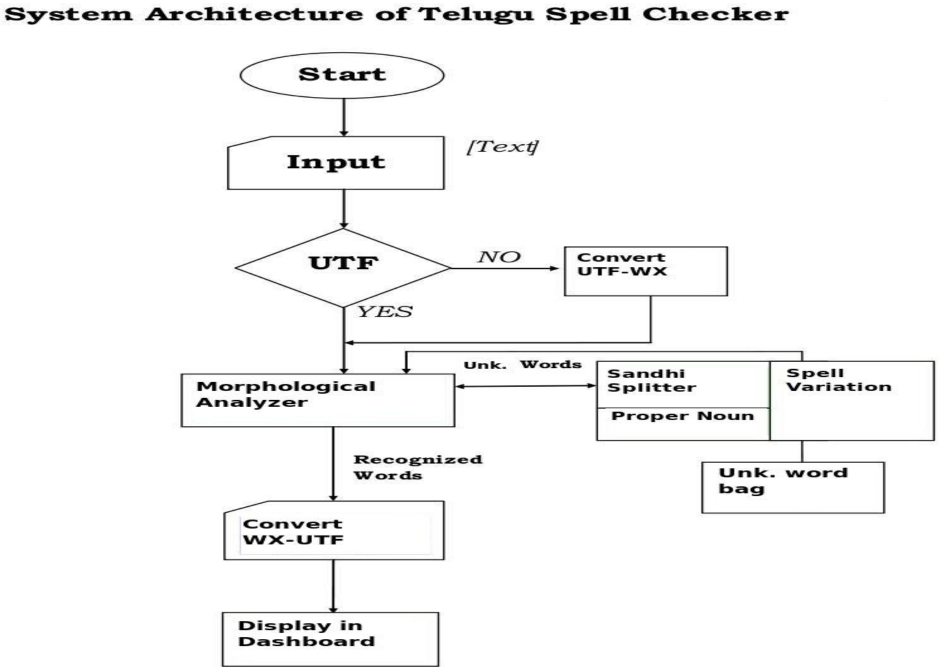

Fig:1

Sandhi Splitter Algorithm :

The approach followed is GENerate-ANAlyze-CONstrain-EVALuate. In this approach, all the possible splits of a given string are first generated and the splits that are not validated by the morphological analyzer are subsequently pruned out.
Currently we apply only two constraints viz.

–C1: All the constituents of a split must be validated by morph.
–C2: All unsegmented words should be validated by spell variation rules.

The Sandhi System flow is presented in Fig: 2

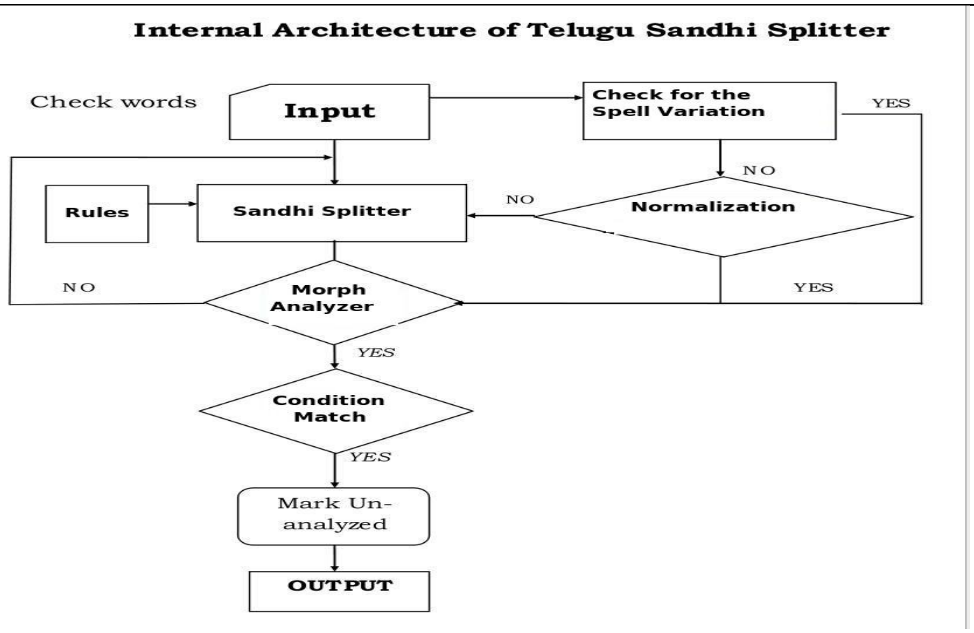

Fig: 2

The basic outline of the algorithm is:

1. Recursively break a word at every possible position applying a sandhi rule and generate all possible candidates for the Input.
2. Pass the constituents of all the candidates through the morph analyser.
3. Declare the candidate as a valid candidate, if all its constitutes are recognised by the morphological analyser and satisfy the conditions that are there in Rule file.
4. No split is possible then, Normalize the input and pass it to the morphological analyser.

WX-Notation used in the Transcription of examples:

a A i I u U q Q eV e E oV o O M H ;
k K g G f c C j J F t T d D N w W x X n p P b B m y r rY l lY v S R s h

Reference:

Bharati, A., Chaitanya, V., Sangal, R.: *Natural language processing: a Paninian perspective*. Prentice Hall of India (1995)

Golding. A.R. 1995. *A Bayesian hybrid method for context- sensitive spelling correction* Proceedings of the Workshop on Very Large Corpora, 1995, 39-53.

Greenberg, J.: 1960. *A quantitative approach to the morphological typology of language*. International Journal of American Linguistics 26. pg. 178–194

Krishnamurti. Bh and J.P.L. Gwynn. 1985. *A Grammar of Modern Telugu*. New Delhi. Oxford University Press.

Mittal, V.: 2010. *Automatic Sanskrit segmentizer using finite state ransducers.* In: Proceedings of the ACL 2010 Student Research Workshop, Association for Com- putational Linguistics. Pg. 85–90

Macdonell, A.A.: 1926. *A Sanskrit Grammar for students*. D.K. Printworld (P) Ltd., New Delhi, India.

Selkirk, E.: 1981. *On prosodic structure and its relation to syntactic structure*. Nordic Prosody II: Papers from a Symposium. Pg. 111–140

Zwicky, A.: 1982. *Stranded to and phonological phrasing in English*. Linguistics 20. Pg. 3–57

Sapir, E.: 1921. Language: *An introduction to the study of speech*. Dover Publications.

Uma Maheshwar Rao G. 1999. *A Morphological Analyzer for Telugu* (electronic form). Hyderabad: University of Hyderabad.

Uma Maheshwar Rao, G. 2002. A Computational Grammar of Telugu. (Mimeo) Hyderabad: University of Hyderabad.

Uma Maheshwar Rao, G. 2005. Telugu Hyper Grammar. (Mimeo and Electronic form) Hyderabad: University of Hyderabad.

(1) Uma Maheswar Rao G, Amba P. Kulkarni and Christopher M. 2007. Functional Specifications of Morphology (mimeo). Hyderabad.