Morphological-based Spellchecker for Sanskrit Sentences

Mrs. Namrata Tapaswi, Dr. Suresh Jain, Mrs. Vaishali Chourey

Abstract : Sanskrit (संस्कृत), called the mother of all Indian languages, plays important role in Indian literature. All the Indian languages are expected to be derived from Sanskrit language. If we change the order of words in formation of the Sentences in Sanskrit, the meaning will remain same i.e., Sanskrit is free ordering language (or syntax free language) and there is no ambiguity in the form of the words even if the order changes. Morphological analysis is a core component of language processing for Indian languages .Complexities involved in spell checking of documents in Sanskrit can be analyzed. We have applied morphological analysis to a large number of words in different parts of speech. A spellchecker based on this analysis has been developed. This paper proposes the architecture of the spellchecker and the spell-checking algorithm based on morphological rules.

Keywords: part of speech, morphology, tagging, verb, noun

1. INTRODUCTION

We can define *Words* in various perspectives such as phonological, morphological, grammatical, lexical, semantic, syntactic, orthographic, sociological and psycho linguistic. Morphologically rich languages are characterized by a large number of morphemes in a single word, where morpheme boundaries are difficult to detect because they are fused together. They are typically free-word ordered, which causes fixed-context systems to be hardly adequate for statistical approaches. The stream of orthographic words that is spellcheckers input is text. The perspectives used for spellcheckers and grammar checkers are different. The former is primarily based on vocabulary, while the latter require grammar rules. Spellcheckers may also use rules to reduce the size of vocabulary. A rule-based approach for spellcheckers is preferred for pan-Indian languages due to their morphological richness. For Indian languages such as Sanskrit and Hindi, dictionaries covering all possible inflections, derivations and compounds obtainable from root words does not exist. Not all Sanskrit words in frequent use are stored in the dictionary. For example each noun can have 3 numbers (वचन / vachana) and 7 cases वभ / vibhakti). So, a noun can have 21 different forms (श द प / shabdarupa) each associating a specific meaning to the noun. For a single noun in Sanskrit, over 100 forms that are either adjectives or adverbs may be possible. Similarly, a verb may exhibit over 250 forms. Morphologically rich languages are characterized by a large number of morphemes in a single word, where morpheme boundaries are difficult to detect because they are fused together. They are typically free-word ordered, which causes fixed-context systems to be hardly adequate for statistical approaches.

1. **Mrs. Namrata Tapaswi**
IES,IPS Academy Indore, MP (India)
namrata.v@rediffmail.com
2. **Dr. Suresh Jain**
KCB Technical Academy,
Indore, MP (India)
suresh.jain@rediffmail.com
3. **Mrs. Vaishali Chourey**
Medi-Caps Institute,
Indore, MP (India)
vaishalichourey@yahoo.com

A morphology based spellchecker has other advantages such as its ability to handle the name-identity problem, i.e. it can absorb new words that are not included in the dictionary. New words may be absorbed by categorizing them into appropriate paradigms. Further, the approach can be drawn upon in building grammar checkers. In the natural language processing one of the methods for spellchecker is morphological rule base. The rule based taggers; this is based on rules, which dictate what tag to be assigned to appropriate words. In the current work, we discuss the architecture and implementation of a rule-based spellchecker for Sanskrit, a major Indian Language. The spellchecker is based on the rules of morphology and the rules of orthography. Morphological rules address word categories and their possible inflections. In the coming section we will discuss issues related to rules of orthography. Morphological issues for various word categories are discussed in Section 3. An Algorithm and frame architecture for spellchecker are provided respectively in Sections 4 and 5, evaluation is described in section 6.

2. LITRETURE REVIEW

Various studies have been done for morphology, Ian Eslick, Hugo Liu described the design and implementation of "langutils," a high-performance natural language toolkit for Common Lisp [2]. Namrata Tapaswi and Dr. Suresh Jain introduced how to morph the Sanskrit sentances[3]. Evangelos Dermatas, George Kokkinakis described stochastic tagger that are able to predict POS of unknown words [4]. Doug Cutting , Julian Kupiec described implementation strategies and optimizations which result in speed high speed operation[6]. Mitchell P. Marcus, Beatrice Santorini and Mary A. Marcinkiewicz described how to constructing one such large annotated corpus--the Penn Treebank [11]. Daniel Gildea and Daniel Jurafsky presented a system for identifying the semantic relationships, or semantic roles, filled by constituents of a sentence within a semantic frame[13]. We qualitatively analyze our results by examining the categorization of several high impact papers. With consultation from prominent researchers and textbook writers in the field, we propose the architecture of the spellchecker and the spell-checking algorithm based on morphological rules.

3. SOME ORTHOGRAPHICAL ISSUES

Sanskrit is written in Devanagari script. It maps the phonemic shape (phonemes and their sequence) of a word to Devanagari symbols through one to one mapping. A spellchecker for Sanskrit has to consider the symbols for 3 व्यंजन *vyanjans* (consonants), 3 स्वर *swaras* (2 vowels, nasalization and aspiration) and 15 *matras* (vowels, nasalization, aspiration and halant markers). Twelve *matras* are used to indicate the presence of a particular vowel at respective position in the phonemic representation of the word. A special *matra* called *halant* represents absence of phoneme '*schwa*' instead of indicating presence of it. *Schwa* is latent in consonantal alphabet. Besides these symbols, over 180 cluster characters, commonly occurring mathematical symbols and punctuation marks are considered. An alphabet represents a phonemic sequence <*consonant, 'schwa'*> [2]. A cluster character may be formed by one of the two sequences <*consonant, alphabet*> and <*consonant, consonant, alphabet*>. Following combinations occur as characters in a written script: an independent *vowel*, an independent *consonant*, an independent cluster character, sequence <*alphabet, matra*> and sequence <cluster character, *matra* except *halan*t>. Valid combinations are defined by the rules of orthography, which in turn are based on etymology [3] and phonemic sequences of words [3]. A spellchecker that considers these factors can automatically reject certain invalid sequences and suggest alternatives or autocorrect some of them [3].The rules of morphology need to capture changes in phonemes. These are represented as transformations of *matras* representing corresponding *vowels*. However, when *vowel schwa* combines with a *consonant*, no separate *matra* appears in the corresponding alphabet. This happens in most encodings used today due to latency of *schwa* in Devanagari. With such encodings, transformations of type (*schwa* -> *matra*) or (*matra* ->*schwa*) cannot be handled directly at encoding level.

For example:

In morphological transformation of word राम (ram) to word रामाः (ramaha) the rule (schwa -> ा) is applied on alphabet म (m). However, in Unicode representation of the word राम (ram), vowel schwa is absent. Similarly,rule (matra ु -> schwa i.e. अ (a)) is applied on alphabet च in transformation of word चुर (chur) to word चोरय (चुर + अय् + अ (churay)), while schwa does not occur in the unicode representation of the word. The spellchecker needs to analyze the word from orthographic point of view by applying the orthographic rules given above. If the ultimate vowel in a word is *schwa*, the penultimate vowel is usually written in its long form. In such cases, after morphological transformations, long penultimate vowel (ू or ी , i.e. U or I) in the root word is transformed to short vowel (ु or ि , i.e. u or i) .

4. RULES OF MORPHOLOGY

Morphological analysis is applied to the categories of nouns, pronouns, adjectives, verbs, adverbs, postpositions, conjunctions and interjections. In Sanskrit, it is convenient to use rules of replacement to capture all types of morphological behavior including those captured in examples given below.

(I) Changes to a word's phonemic shape at the end of the word considering the latent schwa as in transformation of राम (ram) to word रामाः (ramaha) as discussed above.

(II) Changes to a word's phonemic shape not only at the end of the word but anywhere in the middle of the word as in transformation of हरिश्य हरश्च (harishy harshy) to हरिहरौ (hariharou).

(III) Changes to all vowels in the phonemic shape of the word such as in transformations of नरः (narh) to नरौ (naraou).

(IV) Other examples include deletion of ultimate or penultimate consonant, addition of a consonant and vowel pair at the end of the word.

Rules of replacement are generic enough to cover all possibilities of additions and deletions of *consonants* and *vowels*. Replacement rules consider latent *schwa* and null components as and when required. In Sanskrit, postpositions are attached to oblique forms of nominal and verbal entities. Hence, postposition morphology is important for morphological analysis of these categories. Most of the rules can be expressed in the form of transformation tables. Order of suffixes is captured through additional syntactic rules. Over 13,000 root words have been collected and classified by part of speech. For each word category, analysis was performed to derive inflectional morphological rules. Primarily, the parameters that were considered are tense, aspect, mood and gender, number, person and attachment of postpositions.

4.1 Postposition Morphology

Paradigms of postpositions are created based on their linguistic behavior. They include case markers (*vibhakti pratyay*) and a class of postpositions called *shabdayogi avyay*. The latter are attached to singular and plural forms of nouns and pronouns. Some *shabdayogi avyays* exhibit specific behavior. For example, some postpositions need to be written separately when they follow syllable अह (ah), which is a case marker. Some *shabdayogi avyays* can be suffixed with case markers अ (a), औ (aou), अ (aa). Some *shabdayogi avyays* can be composed of others. Postpositions हे (he) and औ (aou) can be attached before some *shabdayogi avyays*, but not before *vibhakti pratyays*. Some *shabdayogi avyays* can be attached to different oblique forms of verbs. Currently, the spellchecker handles the first level of postpositions in the above classification.

4.2 Noun Morphology

In the singular and plural forms of nouns changes due to the attachment of post positions are different. The changed form of a noun to which such attachment is done, is called *Saamaanya roop* (oblique form) of that noun. For example, in morphological transformation of word रामः (ramh) to word रामौ (ramou), the *samanya roop* of रामः (ramh) is रामाः (ramaha).

4.3 Pronoun Morphology

A pronoun has a specific single oblique form to which all *shabdayogi avyays* are attached. We have prepared a list of all possible (over 500) inflections of all pronouns because pronouns show very irregular behavior.

4.4 Verb Morphology

The basis of verb morphology analysis is *Aakhyaata Theory*. It systematically segments the verb forms into verb roots and terminating suffixes called *Aakhyaatas. Aakhyaata* represents information about mood and person. They are named according to the phonemic shape such as *taakhyaata, vaakhyaat* and *laakhyaata*. A regular verb root generates over 100 forms. In addition to regular verbs, there are over 40 irregular verbs.

4.5 Adjective Morphology

Adjectives are classified in inflectional and non-inflectional categories. Inflections result from gender, number and attachment of postpositions to the noun modified by such adjective. Table 1 shows inflectional rules. In the spellchecker, the root form is chosen as masculine form, from which other forms are generated.

Changing part in masculine form	Change		
	Feminine	Neuter	Oblique form
सः (sah)	सा (sa)	तत् (tat)	सः (sah)

Table 1: Adjective Morphology

When genitive case markers or some *Shabdayogi avyays* are attached to nouns, it produces adjectives. These forms are automatically covered in noun morphology.

4.6 Adverb, Conjunction and Interjections

This is an important class of part of speech, for which the rule-based approach proved to be appropriate. Attachment of postpositions to nouns, verbs and pronouns is one of the strategies of adverb formation. In addition, there are non-inflectional adverbs. The set of derived adverbs is automatically covered at the level of morphology of postpositions, nouns, verbs and pronouns. The list of all lexicalized adverbs is constructed. Similarly, all conjunctions and interjections are handled as a list since they are non-inflectional. When some postpositions are attached to demonstrative pronouns, conjunctions are derived. These are handled at the level of rules for pronouns and postpositions.

5. ALGORITHM

Algorithm is designed for checking validity of a word.

1) If the word w is not found as it is in the vocabulary, proceed to step 2, else accept the word and terminate.

2) Scan the word w from right to left to identify a valid suffix string 's2' such that s2 occurs in at least one rule of the form (s1 -> s2). Note that s1 and s2 may be of length more than 1, and s1 may be a substring in s2. If such a rule is not found, reject the word as invalid and terminate, else proceed to step 3.

3) At the rear end of the word, carry a transformation (s2 -> s1) to obtain pruned word w1 from w. If the transformed word w1 is found in vocabulary and if the rule (s1-> s2) is applicable for the word class of w1, accept w as valid word and terminate, else proceed to step 4.

4) Go to step 2 to find another applicable rule.

If the word found as invalid, suggestions are provided based on left to right matching supported by inflectional rules and a string distance. Besides morphological analysis, the spell-checker also considers the rules of orthography as discussed in Section 2. The Spellchecker is implemented in Java. For display, the documents are converted into Unicode.

6. FRAME ARCHITECTURE OF THE SPELLCHECKER

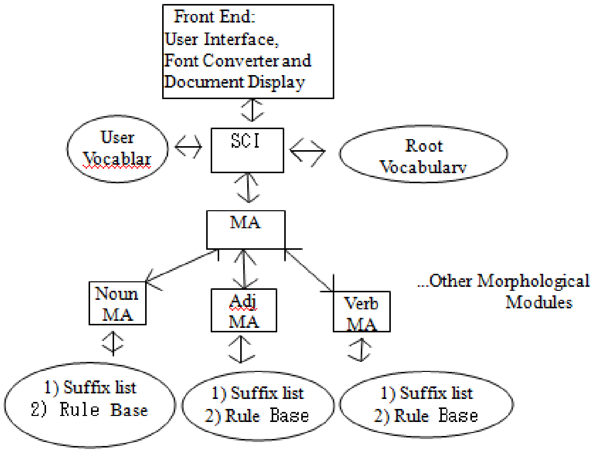

Figure1: Frame Architecture of the spellchecker

Figure 1 shows the frame architecture of the spellchecker. Using the services offered by spell checker's interface (SCI), the front end of the system provides spell checking facilities for Sanskrit documents. A font converter is supported to process convert documents in other formats. Unicode is used for the display unit. The front end provides support for text editing, storage format conversion, highlighting of invalid words and handling of user actions on them. A highlighted word can be ignored, replaced or can be added to user's vocabulary. Alternatives are suggested based on a string distance and morphological rules. The SCI consults the Morphology Analyzer (MA), which in turn consults individual part of speech analyzers for noun, adjectives, verb and other categories. The individual part of speech analyzers use their independent rule bases as shown in the figure 1. Besides, a user level wordlist can also be plugged in.

7. EVALUATION

A manual analysis of 1500 words from a corpus, which were declared by the spellchecker as valid showed that 15 words among them were invalid. This implied an accuracy of validity of 99%. The reasons of error were traced to missing implementation of rules and exceptional cases. Similarly, a manual analysis of words declared as invalid showed that a large percentage of words were wrongly identified as invalid. The reasons were traced mainly to incomplete vocabularies and also to multiple ordered suffixes which have not been handled in the current version. The current size of the vocabulary is limited to about 13,000 words. Enhancement in the vocabulary will improve the accuracy. Various kinds of errors that can occur include misspelled root word and misspelled or inappropriate suffix and wrong order of attachment of multiple suffixes. Suggestions for words found to be incorrect are provided by considering the word's three constituents, which are root, stem forming suffix and case marker or postposition. A right to left (depth first) strategy is used to locate all possible correct formulations. A suggested formulation is allowed to differ at most by one vowel and one consonant. Finally, all suggestions are sorted based on string distance and first eight suggestions are displayed. It was found that in most of the cases that were tested this scheme resulted in obtaining the expected word in first three suggestions if the input word is misspelled by a vowel and/or a consonant.

8. CONCLUSION

Morphological analysis on over 1000 Sanskrit word forms was performed for different part of speech categories. As typical to Indian Languages, the possible inflections of a single word are huge in number. Some challenges in building a spellchecker for handling such complex linguistic phenomenon were discussed. A spellchecker architecture and implementation for first level suffixes based on morphological analysis and rules of orthography was presented. Initial tests showed that the approach was very accurate in declaring words as valid. Further enhancements of derivational morphology will help in increasing the vocabulary. Besides enhancing word lists and rules, enhancements for representing rules for ordering of multiple suffixes in all part of speech categories are required. More elaborate orthographic rules need to be incorporated. Morphology based spellchecker may be extended to include further syntactic and semantic analysis. Besides spellchecking, the morphology based analysis is currently being used in a few applications at the Center for Indian Languages. The morphological analysis of a word serves as a foundation for POS- tagging. Similarly, it is being used in stemming for searching root words in Sanskrit Wordnet.

REFERENCES

[1]D. Jurafsky & J. H. Martin *Speech and Language Processing*. Parson Education

[2]Ian Eslick, Hugo Liu, "Langutils: A Natural Language Toolkit for Common Lisp",

[3] Namrata Tapaswi and Dr. Suresh Jain. "Morphological and Lexical Analysis of the Sanskrit Sentences". MIT International Journal of Computer Science & Information Technology Vol. 1 No. 1 Jan. 2011 pp. 28-31.

[4] Evangelos Dermatas, George Kokkinakis,
" Automatic Stochastic Tagging of Natural Language Texts" , Association for Computational Linguistics" ,1995.

[5]Automatic stochastic tagging of natural language texts by Evangelos Dermatas, George Kokkinakis . MIT Press Cambridge, MA, USA

[6]Doug Cutting , Julian Kupiec , Jan Pedersen , Penelope Sibun, A practical part- of-speech tagger, Proceedings of the third conference on Applied natural language processing, Trento, Italy March 31-April 03, 1992,.

[7]Marie Meteer , Richard Schwartz , Ralph Weischedel, Studies in part of speech labelling, Proceedings of the workshop on Speech and Natural Language, Pacific Grove, California February 19-22, 1991,pp.331-336.

[8]C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge,1999.

[9] E. Charniak *Statistical LanguageLearning*. MIT Press, Cambridge, London,1997.

[10] B. Megyesi, *Improving Brill'S POS Taggerfor an Agglutinative Language*, Stockholm University,1999.

[11]Mitchell P. Marcus, Beatrice Santorini and Mary A. Marcinkiewicz: "Building a large annotated corpus of English: the Penn Treebank": Computational Linguistics, Volume 19, Number 2, 1994,pp.313-330.

[12] Michael Collins: "A New Statistical Parser Based on Bigram Lexical ependencies": Proc. the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 1996, pp.184-191.

[13] Daniel Gildea and Daniel Jurafsky: "Automatic Labeling of Semantic Roles":Computational Linguistics, Volume 28, Number 3,2002, pp. 245-288.