Development of Morphological Tagger for Tamil

Vasu Ranganathan
Language Resource and Research Center
University of Pennsylvania, Philadelphia, USA
vasur@ccat.sas.upenn.edu

Although there have been extensive works on creation of online dictionaries, corpora and word processors for Tamil, there is hardly any work for the development of morphological taggers for Tamil. Despite the complexities that the Tamil words exhibit in general, the literary variety of Tamil is fully manageable, and it is quite possible to develop a robust tagger provided the variety of language is restricted to literary form rather than the dialectal forms. As creation of a successful tagger is a pre-requisite for any natural language processing task, it would imply that the higher level tasks such as building thesauri, spell-checkers, grammar checkers, man-machine interface and so on become less viable without this effort. This paper attempts to describe a Tamil tagger that was built by implementing the principles of the theory of Lexical Phonology and Morphology and is tested with a number of natural language processing tasks including a sample English-Tamil translation system, man-machine communication system and a spell-checker routine. The tagger, written originally in Prolog (http://ccat.sas.upenn.edu/plc/tamilweb/software/tagtamil.zip, tamilnlp.zip) and is in the process of being imported to the web in ASP script, is built with a knowledge base consisting of the rules of morphology of Tamil in a systematic manner in that the processing of input words takes place with suitable consultations of the knowledge base in successive stages.

The two important aspects of any morphological processing component include both recognition and generation of words. The term recognition here means the ability for a computer to recognize and separate affixes from root form of input words, and subsequently the term generation means the ability to generate conceivable word forms in the language by concatenating root form of words and affixes together, after accomplishing necessary sandhi alternations. The agglutinative nature of Tamil language is abundant with complex phonological and morphological rules that undergo during the addition of suffixes into stem. This means that the tagger should be capable of accounting for all such information during the process of recognition and generation.

Three different coding procedures are adopted to recognize a majority of Tamil word forms in their written literary style, and output a list containing information such as type of word, root form of the word and suitable morphological tags for affixes. For example, consider the following simple input and output of this system:

Roman input: eTttukkoNTavarka9aiyaa?
Gloss: 'Is it those who took something? '
Output: [[acc, eTu, pn_refl_hum.pl, inte]]

The tags acc represents the nature of the word which is an accusative noun; pn represents participial noun; refl is meant for reflexive form; hum.pl is for human plural and inte means interrogative.

When a chunk of text is fed to this system, it processes individual sentences and produces a sequence of lists containing information about every word.

Roman input:
mutalvarkaL kuuTTattait toTangki vaitta piratamar vaajpeeyi, teeciya natiniirk koLkaiyai viraivil vakukkavum, natiniirp pangkiiTu toTarpaana piraccinaikaLait tiirkka neRimuRaikaLai uruvaakkavum uRutiyaLittaar.

Tagged Output:

```
["nom","mutalvar","noun","pl"],["acc","kuuTTam","noun"],
["avp","toTangku"],["pa_ajp","vai"],["nom","piratamar"],
["nom","vaajpeeyi"],[""],["pa_ajp","teecu"],["nom","ndatindiir"],
["acc","koLkai","noun"],["loc","viraivu","noun"],
["inf","vaku","conj"],[""],["nom","ndatindiir"],
["nom","pangkiiTu"],["adj","toTarpu","noun"],
["acc","piraccinai","noun","pl"],["inf","tiir"],
["acc","ndeRimuRai","noun","pl"],["inf","uruvaakku","conj"],
["pa","uRutiyaLi","3mas.sg"],["period","."]]
```

Thus, this list representation of input words and sentences is used as a machine recognizable form containing all the information about suffixes in a more explicit manner. Further, the complexities of Tamil words due to sandhi alternations such as occurrence of glides, empty morphemes etc., are discarded retaining all the significant information. This system is capable of recognizing and generating considerable number of Tamil word forms including finite and non-finite form of verbs such as aspectual forms, modal forms, tense forms besides the noun forms such as participial nouns, verbal nouns, case forms and so on.

Despite the complex nature of morphological rules, the order of occurrence of suffixes in Tamil words is quite regular in that the identification every suffix in both and nouns and verbs occur in a finite sequence.

Structure of Tamil noun forms:
Noun + oblique suffix + plural + case + conjunctive suffix + interrogative suffix.

Structure of Tamil verb forms
Verb + tense/infinitive / adverbial participle suffix + aspectual forms
+ (Person, Number and Gender)/relative participle marker + participial forms.

A small-scale dictionary that is built as part of this system contains information about the root form of words and grammatical information describing the nature of words. Different types of verbs are identified by their classification. Following is a list of entries taken from the dictionary that is used as part of sample English-Tamil machine translation system.

```
word("gardener","noun","tooTTakkaarar","aar")
word("so","adv","appaTi","")
word("startle","pr","taTumaaRaccey","1")
word("direct","adj","ndeer","")
```

The type of tagged output illustrated above can be used as the base structure as part of any Natural Language Processing application such as English-Tamil machine translation, Man-Machine interface to be used with any database application, Tamil learning, thesaurus, spell-checking and so on.

As for using the tagged output as part of any NLP application, construction of a syntactic parser is presumed to be significant. A parser is built adopting the theories of Government-Binding (GB) and Context-free Grammar (CFG) produces output as illustrated below.

2) Input:
aracan mandtiriyiTam ivarkaLukku evvaLavu paNam koTukkalaam enRu keeTTaan

"King asked the minister how much money he can give to these poets"

Output from Tagger:

```
[["nom","aracan","noun"], ["hloc","mandtiri","noun"], ["dat","ivarkaL","noun"],
["mass","evvaLavu","ques"], ["nom","paNam","noun"], ["possi","koTu"], ["comp","enRu"],
["pa","keeL","3sgmas"]]
```

Output from CFG parser:

```
[[[["nom","aracan","noun"]]],[[["hloc","mandtiri","noun"]]],[[["dat","ivarkaL","noun"]]],
[[["mass","evvaLavu","ques"], ["nom","paNam","noun"]]], [[["possi","koTu"]]],
[[["comp","enRu"]]], [[["pa","keeL","3sgmas"]]]]
```

This list structure along with a number of list manipulation utilities as part of the programming language PROLOG such as subset, sublist, union etc., are conveniently used to perform various NLP tasks including comparison of sentences within a corpus, search for specific information in a given text, machine understanding of Tamil sentences and so on. To cite one example, following is an example output from English-Tamil machine translation system that translates sentences from Rabindranat Tagore's poem "Gardener". Besides identifying the structure of Tamil words and sentences, this system attempts to identify the structure of English words and sentences using an English parser, and produces the Tamil output using the generation module of the Tamil tagger as discussed earlier.

English input: If you would have it so, I will end my singing

Tamil output: ndiingkaL itai appaTi vaittiruppiirkaL enRaal ndaan ennuTaiya paaTuvatai ndiRuttuveen

English input: If it sets your heart aflutter, I will take away my eyes from your face

Tamil output: itu ungkaLuTaiya itayattai paTapaTakka ceykiRatu enRaal ndaan ennuTaiya kaNkaLai ungkaLuTaiya mukattilirundtu eTuttuviTuveen

English input: if it suddenly startles you in your walk, I will step aside and take another path

Tamil output: itu ungkaLai ungkaLuTaiya ndaTaiyil tiTiirenRu taTumaaRacceykiRatu enRaal ndaan ndakarndtupooy veeRoru vazhiyai eTuttuviTukiReen

The other applications that are tested using this tagger include a Man-Machine interface to interact with both a flat text database as well as relational databases including MS Access and Oracle. Success of this system fully depends upon the availability of a robust text corpus of Tamil prose in literary variety along with a systematic online dictionary with adequate grammatical information and root form of words.

References:

Annamalai, E. (1997). Adjectival Clauses in Tamil. Institute for the Study of Languages and Cultures of Asia and Africa, Tokyo University of Foreign Studies, Tokyo.

Chomsky, Noam. (1982). Lectures on Government and Binding, Foris Publications.

Clocksin, W.F. and C.S. Mellish (1987). Programming in Prolog. (3rd ed.) Springer-Verlag: New York.

Kroch, A., and Joshi, A. (1985). The Linguistic Relevance of Tree Adjoining Grammar. Technical Report MS-CS-85-16., University of Pennsylvania Department of Computer and Information Sciences.

Mohanan, K.P. (1986). The Theory of Lexical Phonology. Dordrecht: Reidel.

Schiffman, Harold (In Print). A Reference Grammar of Spoken Tamil. (Revised edition. Cambridge University Press: Cambridge

Shanmugam, S. V. (1986). collilakkanak kooTpaaTu (in Tamil). Annamalai University: Annamalai Nagar.

Vasu, R. (1993). ``A logical approach to development of natural language understanding system for Tamil." PJDS 3:1,53-64).

__________ (1997a). ``A Lexical Phonology Approach to Processing Tamil Words by Computer." IJDL Vol. XXVI No. 1, January 1997.

__________ (1997b). ``Significance of Creation and Use of Corpus of Modern Tamil Prose Text through the Web." Paper presented in the International Symposium for Tamil Information Processing and Resources on the Internet, National University of Singapore, Singapore, May 1997.