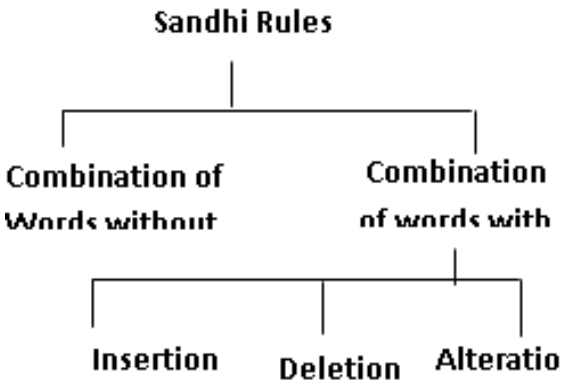
Morphological Analyzer

Textual information available on the web is growing rapidly. This textual data may contain compound words, colloquial words and numerals. These types of words exhibit high degree of inflectional and derivational morphology. This paper describes a rule based approach based on finite state transducer for handling both compound and numerals and a pattern mapping based approach to handle colloquial words. To the best of our knowledge, no previous work has been made to transfer the written colloquial (informal) Tamil form into written normalized formal Tamil form. For this reason, in our approach, we adopt spelling variations rules and perform the mapping for transforming informal written word into formal written word. All compound, colloquial and numeral analyzer have been integrated with the conventional morphological analyzer.

Here, we integrate the conventional morphological analyzer with the compound word, colloquial word and numeral word morphological analyzers. While the compound and numeral morphological analyzers are based on finite state transducers, the colloquial word analyzer uses pattern mapping based approach to get the formal Tamil word. The compound word analyzer handles not only simple compounding but also compounding between two words that may cause inflectional variations during the compounding process.

Finite state transducer approach for compound words

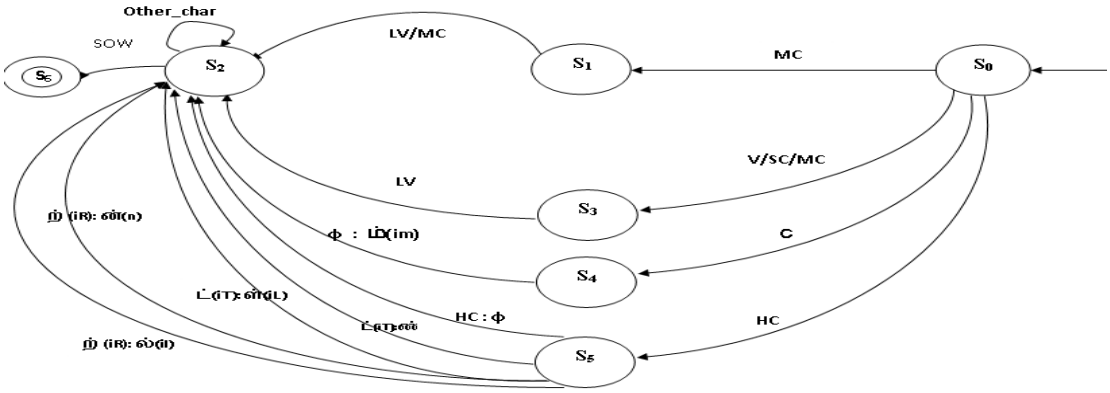
Tamil compound words are formed from root words using sandhi rules which are basically classified into compounding without modifications and compounding with modifications rules. The com-pounding with modification rules can be further classified according to the type of operations such as insertion, deletion, replacement.



Finite State Transducer is a finite state automata with two tapes which describes the input (surface form) and output (lexical form) sequences. It has 7 tuples as given:

- $\Sigma 1$ represents the finite alphabet, namely the input alphabet (a_{i1},.....a_{ik})
- $\Sigma 2$ represents the finite alphabet, namely the output alphabet (b_{i1},.....b_{ik})

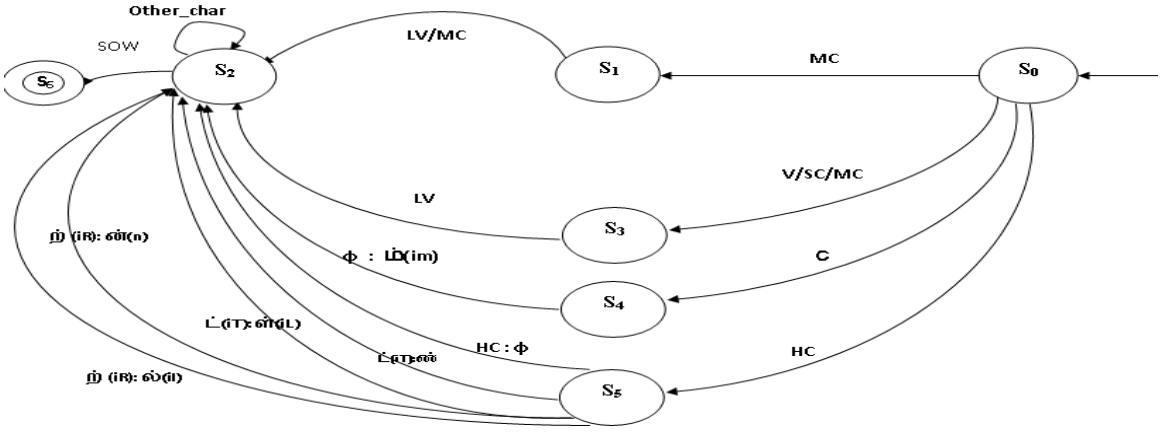
- Q is a finite set of states (S_0 ,S_1,S_2,S_3,S_4, S_5, S_6)
- i ϵ Q is the initial state(S_0)
- F is a subset of Q, the set of final states; (S_6)
- Here *a:b* represents the replacement of *a* in the surface form to *b* in the lexical form
- *c/d* states that transition can occur if either *c* or *d* is in lexical form.



Finite state Transducer (FST) for Compound Analyzer

For compound analyzer the finite set of states Q has 7 states from S_0 to S_6. Rule 1 describes simple compounding where no modification occurs. When the two root words join, however two cases exist for rules of category A, where S_0-S_1-S_2 -S_6 is the rule that describes transition when both constituents are consonants. The other case S_0-S_3-S_2-S_6 describes the transition when first constituent ends with consonant and the second constituent starts with vowel. All category of Rule B involves morphological modification, when the two root words join. Hence S_0-S_4-S_2-S_6 gives an example of the insertion of an alphabet to the lexical form, to form the word corresponding to the first constituent. S_0-S_5-S_2-S_6 explains the second type of rule in category B. Here an alphabet at the surface level is replaced by another alphabet at the lexical level. The final type of rule category B is corresponds to S_0-S_5-S_2-S_6, where an alphabet of the surface form gets deleted.

Finite state transducer approach for numeral analyzer



Finite state Transducer (FST) for Numeral Analyzer

In the S_O-S_1-S_2-S_6 type of rule, one alphabet is added to the lexical form. In S_O-S_3-S_2-S_6 type of rule, two alphabet are added. In S_O-S_3-S_2-S_6 type of rule deletion of an alphabet from the surface form takes place. In rules of category D, a single alphabet in the surface form may be replaced by a single alphabet (S_O-S_1-S_2-S_6) or two alphabets (S_O-S_4-S_2-S_6) in the lexical form. In rules of category D, two alphabets in the surface form may be replaced by a single alphabet (S_O-S_1-S_2-S_6) or two alphabets (S_O-S_1-S_2-S_6) in the lexical form.

After morphological analysis of compound words and numerals, if a failure still occurs, it is either due to absence of constituent words in the dictionary or because morphological suffixes could not be stripped from the original word. This failure to remove morphological suffixes from words could be because the word is in the colloquial form.

Pattern based approach for Colloquial Analyzer

The colloquial analyzer processes the word by scanning from right to left character by character where character is (consonant + vowel -> **க்**+**அ** = **க**). The pattern based approach based on spelling variation rules is used to convert colloquial form of word to its normal form. This normal form can then be processed by the conventional morphological analyzer to identify its root and morphological suffix**.**

The pattern based approach basically converts the pattern P1of the colloquial form to a pattern P2 of the normal form. The pattern P1 normally occurs as morphological suffixes at the end of the word or in some cases it may occur anywhere in the colloquial form of the word. Some changes of P1 to P2 is independent of the context in which P1 occurs while sometimes the change from P1 to P2 is decided by one or two characters that occurs in the preceding context of P1.

P1: Pattern to be replaced

P2: Replacing pattern

X: Preceding character

Y: Preceding to the preceding character

The first four tables represent the mapping of the suffix (endings) of the colloquial word. Table 1.1.1 shows patterns P1 that can be directly mapped to P2. Table 1.1.2, lists patterns P1 which are changed to P2 and then need to undergo morphographemic changes from P2 to P3 to get the formal Tamil word. Table 1.1.3 lists patterns that can be mapped to P2 only if the preceding character is X. Table 1.1.4, lists patterns P1 that can be mapped to P2 only if the preceding characters are X and Y. Table 1.1.5 indicates patterns P1 that occur anywhere in the word which are mapped to P2.

Table 1.1.1 – Suffix Mapping of ending patterns

P1	P2	Example
ங்க(nga)	ர்கள்(rgal)	வருவாங்க (varuvaanga) – வருவார்கள் (varuvaargal)

இயா(iyaa)	ஆயா(Ayaa)	பார்த்தியா (paarthiya) -பார்த்தாயா(paarthaya)
ணும்(num)	வேண்டும் (Vendum)	பார்க்கணும் (paarkanum) – பார்க்கவேண்டும் (paarka Vendumm)
கிற(kira)	கிறாய்(kiRaai)	படிக்கிற (padikira) - படிக்கிறாய் (padikiRaai)
ற(Ra)	கிற (kiRa)	ஒடுற (oduRa) - ஒடுகிற (odukiRa)
உன(una)	இன (ina)	ஒடுன (oduna) - ஒடின(odina)
ப்புறம்(ippuram)	பிறகு(piragu)	படிச்சப்புறம் (padichchaippuram)= படிச்ச பிறகு (padichcha piragu)
ன்னா (nnaa)	என்றால் (enRaal)	தப்புன்னா (thappunna) - தப்பு என்றால் (thappu enRaal)
கேன் (Ken)	கிறேன் (kiRen)	இருக்கேன் (iruKen) - இருக்கிறேன் (irukiRen)
கீங்க (Kiinga)	கிறீர்கள்(kiReergal)	வந்திருக்கீங்க (vandhiruKiinga) – வந்திருக்கிறீர்கள் (vandhirukkiReergal)

Table 1.1.2 – Suffix Mapping of **ending patterns** with Morphographemic changes

P1	P2	Example	Morphographemic change (P3)
கிட்டே (kittey)	இடம் (idam)	தம்பிகிட்டே (thambikittey)- தம்பிஇடம் (thambi yidam)	தம்பி + ய் +இடம் = தம்பியிடம் (thambiyidam)
வோட (voda)	உடைய (udaiya)	அம்மாவோட (ammavoda) - அம்மாஉடைய (amma - vudaiya)	அம்மா + வ்+ உடைய = அம்மாவுடைய (ammavudaiya)
லே (Ley)	இல் (il)	கடைலே (kadaiLey) - கடைஇல் (kadai - yil)	கடை+ ய்+ இல் = கடையில் (kadaiyil)
ஓடே (odey)	ஓடு (odu)	தம்பியோடே (thambiyodey) – தம்பிஓடு (thambi - yodu)	தம்பி + ய் +ஓடு = தம்பியோடு (thambiyodu)

Table 1.1.3 – Suffix mapping of **ending patterns** with checking of one preceding character

X	P1	P2	Example
ஆ (A) / ஏ (E)	ஞ்சு (nju)	ய்ந்து (yndhu)	காஞ்சு (kaanju) - காய்ந்து (kaaindhu)
இ (i)	ஞ்சு (nju)	ந்து (ndhu)	நொடிஞ்சு (nodinju) – நொடிந்து (nodindhu)
எ (e)	ஞ்சு (nju)	ய்து (ydhu)	செஞ்சு (senju) – செய்து (seidhu)
க் (k) / ம் (m) / ந் (n) / ச் (ch) / ஏ (E)	த்து (ththu)	ற்று (RRu)	காத்து (kaaththu) – காற்று (kaatRRu)
ஓ (O) / ஆ (A)	ச்சு (chchu)	யிற்று (yitRu)	போச்சு (poochchu) – போயிற்று (pooyitRu)
இ (i) / ஐ (ai)	ச்சு (chchu)	த்து (ththu)	பிடிச்சு (pidichchu) – பிடித்து (pidiththu)
இ (i)	ச்ச (chch)	த்த (thth)	நடிச்ச (nadichcha) – நடித்த (nadiththa)
ஐ (ai) / இ (i) / உ (u) / ர் (r)	ல (la)	இல் (il)	நேரத்துல (neraththula) – நேரத்தில் (neraththil)
அ (a)	ல (la)	இல்லை (illai)	போகல - போகவில்லை

Table 1.1.4 – Suffix mapping of **ending patterns** with checking of two preceding characters

Y	X	P1	P2	Example
ப் (p)	ஆ (A)	த்து (ththu)	ர்த்து (rththu)	பாத்து (paaththu) - பார்த்து (paarththu)
அ (a) / எ (e) / இ (i)	ப் (p)	ப (pa)	பொழுது (pozhudhu)	அப்ப (app) – அப்பொழுது (appozhudhu)

Table 1.1.5 – Suffix mapping of patterns occurring at any place

P1	P2	Example
ன்ன (nn) / **ண்ண** (NN)	**ன்ற்** (nR)	**இன்னைக்கு** (innaikku)- **இன்றைக்கு** (inRaikku)
கிட்டு (kittu)	**கொண்டு** (kondu)	**வந்துக்கிட்டுருக்கிறான்** (vandhukkitturukkiraan) - **வந்துக்கொண்டிருக்கிறான்** (vandhukkondirukkiraan)
உடு (udu)	**விடு** (vidu)	**வந்துடுறான்** (vandhuduRaan) – **வந்துவிடுகிறான்** (vandhuvidukiRaan)
உரு (uru)	**இரு** (iru)	**வந்துருக்கான்** (vandurukkaan) - **வந்திருக்கான்** (vandirukkaan)
கிடு (kidu)	**கொள்** (kol)	**இருந்துக்கிடுக்கிறான்** (irundhukkidukkiRaan) – **இருந்துக்கொள்க்கிறான்** (irundhukkolkkiRaan)
க்கிறு (kkiRu)	**க்கின்று** (kkinRu)	**மறைக்கிற** (markkiRa) - **மறைக்கின்றான்** (maraikkiRaan)

Flow diagram for colloquial analyzer