6. A Transfer Module in Tamil – English MT

In the present chapter, it is attempted to design a Transfer module to translate Tamil inflectional categories and their properties into English Equivalences.

Based on the preceding two chapters (ch.4 and ch.5), the necessary Transfer Rules are being proposed to be incorporated in this model.

The computational program has been developed in .net platform. The programming language is C#.

The proposed module would help the Language technologists involved in the development of Tamil – English Machine Translation system.

The proposed module comprises of many sub-modules. These sub-modules are explained here in detail.

At the end of the chapter, the sample program has been explained with the necessary screen shots to demonstrate the importance of the study of the Inflectional categories and their properties in both Tamil and English.
6.1. Structure of Transfer Module

The automatic translation of Tamil inflected wordforms in a sentence into English equivalences involves many processes. They are:

1. Tokenization
2. Morphological Parsing
3. POS Tagging
4. Clitics Filtration
5. Derivation of Stem
6. Transfer of Tamil wordforms into English Equivalences

During the above processes, at some points, some decisions may have to be made. They are:

1. After Morphological Parsing and POS Tagging, it has to be decided whether the output is a Variable Lexeme or not. For example, some determiners such as anta, oru are not variable lexemes. Since the present Transfer module is concerned only with the Variable one, this decision-making is quite important at this stage.

2. After this, it has to be decided whether the output is inflectional wordform or not. The variable output may be a derived one such as alakāṇa ‘beautiful’. These derivations having no inflectional suffixes should be discarded at this stage for further process.

3. The above inflected wordform may have some clitics. For example, in paṭittāṇā ‘did he study’, the final morphemic segment is a clitic -ā. Since the aim of the Transfer module in the
present work is to translate the inflectional properties only, any clitic that come across, should be dropped. So, it has to be decided at this stage whether the above output has any clitic.

4. After the process of dropping of clitics above, some of the output may be some derived stems. For example, in *alakāṇavaṇai* ‘handsome person (acc.)’, though the final and its preceding morphemic segments are inflectional morphemes, the stem with which they are attached is a derived adjective (*aḷaku+āṇa*) Since the meaning of the derived stem may not be available in the Lexicon, the derivational stem should be parsed to get the meanings of the root and stem and the meaning of the total derived stem.

All the above processes and decision-making are explained in the following flow chat:
The present work is to translate the inflectional properties only; any clitic that comes across should be dropped. So, it has to be decided at this stage whether the above output has any clitic.

4. After the process of dropping of clitics above, some of the output may be some derived stems. For example, in "ḻakāṉavaṉai" (handsome person (acc.)), though the final and its preceding morphemic segments are inflectional morphemes, the stem with which they are attached is a derived adjective (aḻaku+āṉa). Since the meaning of the derived stem may not be available in the Lexicon, the derivational stem should be parsed to get the meanings of the root and stem and the meaning of the total derived stem.

All the above processes and decision-making are explained in the following flow chart:

Figure 6.1 Structure of Transfer Module in Tamil – English MT
6.1.1. Input

The input to the process may be any sentence type – Simple, Compound and Complex, which are grammatical ones. That is, the sentence may be a single clause or more than one clause but it should be a grammatically correct one. But the input should be a single sentence.

e.g.

1. *nāṉ anta māṇavaraip pārtēṉ. ‘I saw that student’*
2. *nāṉ anta māṇavaraip pārkkavum pēcavum ceytēṉ. ‘I saw the student and spoke to him’*
3. *nāṉ anta māṇavaraip pārttu, pēciṉēṉ. ‘Having seen that student, I spoke to him’*

(1) is a simple sentence; (2) a compound sentence; (3) a complex sentence.

Moreover, at this stage, the given sentence should have been normalized for further tokenizing process. For example, in the following input,

*nāṉ avaraik kinarriliruntu tūkki viṭṭēṉ,*

there is a space between the fourth and fifth words. Because of this, the program would consider these words as two separate words (main verbs) – *tūkki ‘having lifted’ and viṭṭēṉ ‘I gave up’. So, the meaning of the sentence would be ‘Having lifted him from the well, I gave up’.

On the other hand, if there is no space between the above words, it will be *tūkkiviṭṭēṉ*. The first part of this wordform *tūkki* would be considered as the verbal participle form of the main verb *tūkku* and the
second part *vittēy* the aspectual auxiliary verb. The resultant meaning of this wordform would be ‘I had lifted him from the well’.

There are problems with clitics, postpositions, adverbial particles also.

1. *uṇkaḷukku varuvatu tāṇē varum.*
2. *uṇkaḷukku varuvatuttaṇē varum.*

In the above example (1), where there is a space between second and third word, the third word *tāṇē* means ‘automatically’ (adverb) or ‘itself’ (adverb).

In (2), since there is no space between the second and third word, the whole unit stands as a single wordform and its meaning is ‘which might come’. Thus, the meanings of the sentences are as follows:

1. *uṇkaḷukku varuvatu tāṇē varum.*
   ‘The one which would come to you would come automatically.’
   ‘The one which would come to you itself will come.’
2. *uṇkaḷukku varuvatuttaṇē varum.*
   ‘That which could come to you only will come. Isn’t?’

In the following example,

1. *avar enatu kaiyaip purrip pēciṉār* ‘Holding my hand, he talked.’
2. *avar enatu kaiyappurrip pēciṉār* ‘He talked about my hand.’

The meaning difference is due to whether there is a space between *kaiyai* and *parri* or not. If there is a space, *parri* is the verbal participle form of the verb *parru* ‘to hold’; when there is no space, it is the accusative postposition (*aipparri*) attached to the noun *kai* ‘hand’.

Another example,
1. uṅkaḷukkup piṭitta vaṇṇam
2. uṅkaḷukkup piṭittavaṇṇam

In (1), the second word is the relative participle form of piṭi ‘like’ and the third word means ‘colour’. So the meaning of the whole phrase is ‘the colour you like’.

In (2), there is no space as in the first phrase. The second word consists of the Relative Participle form of piṭi ‘like’ and the third word is the Adverbial Particle vaṇṇam ‘as per’. So the meaning of this phrase is ‘as you like’.

Here, it is to be mentioned, that Sandhi (morphophonemics) is also to be considered in this normalization process. Depending upon the sandhi, meaning changes as in the following examples:

1. avaṅ kattikonṭu veṭṭīnāṅ ‘He cut with the knife.’
2. avaṅ kattikkonṭu veṭṭīnāṅ ‘He was shouting while he cut.’

In (1), the second word consists of katti ‘knife (noun)’ and konṭu ‘with (postposition)’. The meaning is ‘with the knife’.

In (2), between katti and konṭu, there is an internal sandhi ‘k’. Because of this, here, konṭu stands for the progressive aspectual marker. And the stem katti is the verbal participle form of kattu ‘shout’.

The above examples clearly establish that the normalization process of the input data is very much essential before they are sent to
any analysis by the computer. Otherwise, it would lead to many ambiguities which could not be solved automatically by the computer.

In the present program, it is assumed that the input data are normalized ones.

6.1.2. **Tokenizer**

Here, the *token* stands for the individual wordform. The task of the tokenizer is to identify the tokens (wordforms) available in the sentence.

After receiving the input, the tokenizer would analyse it and provide the individual tokens. Here, the space between the words is the only criterion for this process. If there is any punctuation marker, it would be discarded during this process.

The output of this tokenizing process is the input to the next process – Morphological Parsing.

6.1.3. **Morphological Parser**

The task of this Parser is to segment the individual morphemes of every word, if they contain more than one morpheme.

As Tamil is an agglutinative language, all the grammatical affixes attached to the root/stem of Tamil are suffixes only. The only exception is the demonstrative bound forms – *a-*, *i-*, *e-* found in the words like *ikkulantai* ‘this child’, *akkulantai* ‘that child’, *ekkulantai* ‘which child’. The present parser could identify this prefix type demonstrative morphemes.
There are two ways to segment the Tamil words: one is, from left to right; another is right to left. For Tamil, if the first one is adopted, then the beginning root/stem would be first segmented and then one by one other suffixes would be identified and segmented.

For example, if the input to the parser is the word *paṭittāṅ* ‘studied(he)’, the root *paṭi* would first be identified, then the next past tense suffix *tt* and finally the PNG suffix *āṅ* would be identified and the output would be three parsed morphemes – *paṭi, tt and āṅ*.

If the parsing starts from right, that is from the end of the word, only after getting the suffixes, the root/stem would be identified. That is, with the above example, first the PNG suffix, then the past tense suffix and finally the root morpheme would be identified. However, the final output would be the same - *paṭi, tt and āṅ*.

**Right to Left Parsing**: In the parsing process from right to left, first the whole word would be first checked with the root word database –‘Lexicon’ – whether it is a root word or not. If it is not found in the database, then the process would start to segment the final suffix. In the present example, the PNG suffix *āṅ* would be parsed out. Then, again the remaining segment *paṭitt-* has to be checked with the database whether it is a root or not. If it is not found there, then the parsing process would start to identify and segment the past tense suffix *-tt*-. After its segmentation, once again the remaining *paṭi* has to be checked with the database to decide whether it is a root or not. In the present example, it would be found in the database. Now the parsing process would come to an end.
**Left to Right Parsing**: If the process starts from left to right, first the whole input would be checked with the database to decide whether it is a root or not. If it is not found in the database, then parser, with the help of its algorithm, would attempt to identify the root. Once the root is identified, then for further segmentation process for other suffixes, it would not be necessary to check the remaining segment again and again with the root word database. With the present example, once the segment _paṭi_ is identified as the root, then it would not be necessary to check the remaining segment -_ttāṉ_ or after segmenting the tense suffix -_tt-_ , the final remaining segment _āṉ_ against the root word database.

So, in the present work, ‘left to right’ way is adopted. That is, first the root would be identified and then other suffixes would be parsed.

The Morphological Parser consists of the following components:

1. Root word Database
2. Suffix Database
3. Morphotactics of Tamil wordforms
4. Tamil Sandhi (both internal and external Sandhi)

### 6.1.3.1. Root Word Database structure:

Tamil Root words (lexicons) are stored with the necessary details along with English meanings.

The root words are classified into five major categories:

1. Noun
2. Verb
3. Adjective
4. Adverb
5. *Itaiccol* (‘middle words’)
For verbs, their tense conjugation details are given in the database. Based on the tense suffixes, the Tamil verbs are classified into various groups by grammarians and linguists. There are variations in their classification. For the present work, the conjugation system adopted in the Tamil Lexicon (University of Madras, 1982) is adopted. In that system, there are thirteen classes. The irregular verbs are accommodated in the 13th class.

6.1.3.2. Suffix Database structure

All the inflectional suffixes identified in the Chapter 4 of the present work are put in the database. In addition to that, for the purpose of parsing, the derivational suffixes are also included here. They are classified into Noun suffixes, Verb suffixes, Adjectival suffixes and Adverbial suffixes. *Itaiccol* won’t take any suffix.

**Noun suffixes:**

1. Number (Plural) suffix
2. Fillers
3. Case
4. Postpositions
5. Verbalizers
6. Adjectival suffixes
7. Adverbial suffixes
8. Relative Participle of some of the postpositions
9. NG (Number – Gender) suffixes

The above first four types of suffixes are inflectional suffixes and they are already dealt with in detail in the Chapter 4.
**Verbalizers:** The input to the Parser may be some wordforms consist of a noun plus some verbalizers.

e.g.  *talaivarāṟār* ‘became a leader’

In this example, the morphological parser would identify the segment *talaivar* as a root noun with the help of the root word database. However, this root noun is here followed by a verbalizer suffix ā (it is not the interrogative clitic ). After taking this suffix, the root noun *talaivar* becomes a verb – *talaivarā*. After this, the derived segment *talaivarā* would undergo verb inflection. In this example, this segment takes a past tense suffix ṉ with one more suffix – PNG suffix – ār.

Because of this above phenomenon, all the verbalizer suffixes (ā, āku, ākku) which are added to the noun are identified and kept in this database.

**Adjectival suffixes:**

Some of the wordforms such as *aḻakāṉavaṉai* ‘the handsome person(acc.)’ are noun wordforms. However, in the above example, the noun stem aḻakāṉavaṉ is not available in the root noun database. Because, this is a derived noun, derived from noun root (*aḻaku*) + adjectival suffix (*āṉa*) + NG suffix (*aṅ*). Only after getting this derived noun wordform, the accusative case suffix is added and the final wordform *aḻakāṉavaṉai* is arrived at.

In the parsing process of the above example, since the adjectival suffixes and NG suffixes are involved, these suffixes should be available in the noun suffix database.
Adverbial suffixes:

In Tamil, along with the pure adverb wordforms, there are many derived adverb wordforms such as *aḷakākattāṇ, vēkamākavā*. Though adverbs are variable lexemes (by taking clitics), they do not undergo any inflectional process. Here, it is to be mentioned that clitics are not inflectional suffixes.

The above mentioned derived adverbs are derived from nouns by taking adverbial suffix – āka.

1. *aḷakākattāṇ = aḷaku (Noun)+ āka (Adv. suffix) + (t)+ tāṇ (Clitic)*
2. *vēkamākavā = vēkam (Noun) + āka (Adv. suffix) +(v)+ ā (Clitic)*

Unless the above wordforms are parsed and tagged as adverbs, they could not be determined later whether they are inflected wordforms or not. This knowledge is essential for the present work, since here the aim is to translate the inflectional properties only. Hence the need for including adverbial suffixes along with other suffixes in the noun suffix database.

**Relative Participle form of postpositions**

In fact, the postpositions generally do not undergo inflection. However, some of the Tamil postpositions take relative participle forms. There may be some historical reasons behind this which are to be analysed. But this is beyond the aim of this present work.

e.g. *avanaippaṟri* ‘about him’ = *avan* ‘he’ + *aippaṟri* ‘about’

The above wordform could be followed by some verbs.
Adverbial suffixes:

In Tamil, along with the pure adverb wordforms, there are many derived adverb wordforms such as aḻakākattāṉ, vēkamākavā. Though adverbs are variable lexemes (by taking clitics), they do not undergo any inflectional process. Here, it is to be mentioned that clitics are not inflectional suffixes.

The above mentioned derived adverbs are derived from nouns by taking adverbial suffix –āka.

1. aḻakākattāṉ = aḻaku (Noun) + āka (Adv. suffix) + (t) + tāṉ (Clitic)
2. vēkamākavā = vēkam (Noun) + āka (Adv. suffix) + (v) + ā (Clitic)

Unless the above wordforms are parsed and tagged as adverbs, they could not be determined later whether they are inflected wordforms or not. This knowledge is essential for the present work, since here the aim is to translate the inflectional properties only. Hence the need for including adverbial suffixes along with other suffixes in the noun suffix database.

Relative Participle form of postpositions

In fact, the postpositions generally do not undergo inflection. However, some of the Tamil postpositions take relative participle forms. There may be some historical reasons behind this which are to be analysed. But this is beyond the aim of this present work.

e.g. avaṉaippaṟṟiṉ “about him”

The above wordform could be followed by some verbs.

e.g. avaṉaippaṟṟiṉ pēciṉēṉ “I talked about him”.

The above postposition inflected noun wordform is further inflected for the relative participle suffix -a and the resultant form is:

avaṉaippaṛṛiya = avan + aippari + (y) a ‘about him’

Since the above form is relative participle, it could be followed by noun phrases only.

avaṉaippaṛṛiya ceyti ‘the news about him’

So, it is necessary to have the knowledge of the relative participle suffix in noun wordform parsing.

NG suffixes: It is already explained in the Chapter 4.

Verb suffixes:

1. Tense suffixes (including negative)
2. Voice suffixes
3. Aspectual suffixes
4. Modal suffixes
5. Verbal participle suffixes
6. Adjectival participle suffixes
7. Adverbial participle suffixes
8. Relative Participle suffixes
9. PNG suffixes
10. Verbal noun suffixes
11. Participial noun suffixes

All the above suffixes are already dealt with in detail in the Chapter 4 (ch.4).
6.1.3.3. Morphotactics

Noun wordform

<table>
<thead>
<tr>
<th>Noun+(Plural)+(Filler)+(Case)+(Postposition)+(RP)+ (NG) + (Clitics)</th>
</tr>
</thead>
</table>

e.g.

- Noun - *paiyan* ‘boy’
- Noun + Plural - *paiyankal* (*paiyan + kal*) ‘boys’
- Noun + Case - *paiyanai* (*paiyan + ai*) ‘boy (acc.)’
- Noun + Plural + Case - *paiyankalai* (*paiyan + kal + ai*) ‘boys (acc.)’
- Noun + Case + Postposition - *paiyanaiapparri* (*paiyan + ai +(p)+ parri*) ‘about the boy’
- Noun + Plural + Case + Postposition - *paiyankalaiapparri* (*paiyan + kal + ai +(p) + parri*) ‘about the boys’
- Noun + Postposition - *paiyanmitu* (*paiyan + mitu*)
  ‘on/above/over the boy’
- Noun + Plural + Postposition - *paiyankalmitu* (*paiyan + kal + mitu*) ‘on/above/over the boys’
- Noun + Case + Postposition + RP – *paiyanaiapparriya* (*paiyan + ai +(p)+parri +(y)+ a*) ‘about the boy’
- Noun + Plural + Case + Postposition + RP - *paiyankalaiapparriya* (*paiyan + kal + ai +(p)+ parri +(y)+ a*) ‘about the boys’
- Noun + Case + Postposition + RP + NG - *paiyanaiapparriyatu* (*paiyan + ai +(p) + parri +(y)+ a + (a)tu*) ‘(It is) about the boy.’
- Noun + Plural + Case + Postposition + RP + NG - *paiyankalaiapparriyatu* (*paiyan + kal + ai +(p)+ parri +(y)+ a + (a)tu*) ‘(It is) about the boys.’
- Noun + Filler + Case - *marattai* (*maram + attu + ai*) ‘tree (acc.)’
Noun + Filler + Case + Postposition - *marattaipparri*  
(\(maram + attu + ai + (p) + parri\)) ‘about the tree’

Noun + Filler + Case + Postposition + RP - *marattaipparriya*  
(\(maram + attu + ai + (p) + parri + (y) + a\)) ‘about the tree’

Noun + Filler + Case + Postposition + RP + NG -  
*marattaipparriyatu* (\(maram + attu + ai + (p) + parri + (y) + a + (a)tu\)) ‘(It is) about the tree’

---

**Noun + (Plural) + Verbalizer**

E.g.

Noun + Verbalizer - *talaivarāṅar* (\(talaivar + ā + ṉ\) + ār) ‘became a leader – he’

Noun + (Plural) + Verbalizer - *talaivarkaḷāṅarikaḷ* (\(talaivar + kaḷ + ā + ṉ\) + ārkaḷ) ‘became leaders – they’

---

**Noun + Adjectival suffix + (NG) + ……**

E.g.

Noun + Adjectival suffix - *panpuḷḷa* (\(panpu + uḷḷa\)) ‘virtuous’

Noun + Adjectival suffix + NG - *panpuḷḷavar* (\(panpu + uḷḷa + \text{var}\)) ‘virtuous person’

---

**Noun + Adverbial suffix + (Clitics)**

E.g.

Noun + Adverbial suffix - *aḻakāka* (\(aḻaku + āka\)) ‘beautifully’

Noun + Adverbial suffix + Clitics - *aḻakākattāṉ*  
(\(aḻaku + āka + (t) + tāṉ\)) ‘good only’
**Verb Wordform**

<table>
<thead>
<tr>
<th>Verb + VP-Simple + ASP + (Tense + {PNG /RP + (NG)}) ) / (VP) + (Clitics)</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g.</td>
</tr>
<tr>
<td>- Verb + Tense + PNG - <em>paṭittāṉ</em> (<em>paṇi + tt + āṉ</em>) ‘read/studied-he’</td>
</tr>
<tr>
<td>- Verb + VP-Simple + ASP - <em>paṭittuviṭu</em> (<em>paṇi + ttu + viṭu</em>) ‘have read’</td>
</tr>
<tr>
<td>- Verb + VP-Simple + ASP + Tense + PNG - <em>paṭittuviṭṭāṉ</em> (paṇi + ttu + viṭu + ṭ + āṉ) ‘has read – he’</td>
</tr>
<tr>
<td>- Verb + VP-Simple + ASP + Tense + RP - <em>paṭittuviṭṭa</em> (paṇi + ttu + viṭu + ṭ + a) ‘who has read’</td>
</tr>
<tr>
<td>- Verb + VP-Simple + ASP + Tense + RP + NG - <em>paṭittuviṭṭavaṉ</em> (paṇi + ttu + viṭu + ṭ + a + vaṉ) ‘one who has read’</td>
</tr>
<tr>
<td>- Verb + VP-Simple + ASP + Vpinfi. - <em>paṭittuviṭa</em> (paṇi + ttu + viṭu + a) ‘have to read’</td>
</tr>
<tr>
<td>- Verb + VP-Simple + ASP + VP-Simple - <em>paṭittuviṭṭu</em> (paṇi + ttu + viṭu + ṭ + tu) ‘having read’</td>
</tr>
</tbody>
</table>

**Verb +(Filler) + VP-inf+ (Voice) + (Modal) + (Tense+ PNG)+ (Clitics)**

<table>
<thead>
<tr>
<th>e.g.</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Verb + Filler + VP-inf + Voice - <em>paṭikkavai</em> (paṇi + kk + a + vai) ‘make one study/read’</td>
</tr>
<tr>
<td>- Verb + Filler + VP-inf + Modal - <em>paṭikkalām</em> (paṇi + kk + a + lām) ‘may read’</td>
</tr>
<tr>
<td>- Verb + Filler + VP-inf + Voice + Tense+ PNG - <em>paṭikkapattaṉu</em> (paṇi + kk + a + (p) + paṭu + ṭ + atu) ‘was read-it’</td>
</tr>
<tr>
<td>- Verb + Filler + VP-inf + Modal + Tense+ PNG - <em>paṭikkamuṭṭiṉu</em></td>
</tr>
</tbody>
</table>
Verb + (Filler) + Tense / Negative + RP + (NG)/( Adverbial Particle)

- Verb + Tense + RP - paṭitta (paṭi + tt + a) ‘who read/which was read’
- Verb + Filler + Negative + RP - paṭikkāta (paṭi + kk +āt + a) ‘who did not read/which was not read’
- Verb + Tense + RP + NG - paṭittavaṇ (paṭi + tt + a + vaṇ) ‘one who studied/read’
- Verb + Filler + Negative + RP + NG - paṭikkātavaṇ (paṭi + kk +āt + a + vaṇ) ‘one who did not study/read’
- Verb + Tense + RP + Adverbial Particle - paṭittapōtu (paṭi + tt + a + pōtu) ‘when/during studying/reading’
- Verb + Filler + Negative + RP + Adverbial Particle - paṭikkātavarai (paṭi + kk +āt + a + varai) ‘till one not read’

Verb + (Filler)/Tense + VP + (Adjectival particle)+ Clitics

- Verb + (Filler) + VP-inf - paṭikka (paṭi + kk + a) ‘to read’
- Verb + VP-Simple - paṭittu (paṭi + ttu) ‘having read’
- Verb + (Filler) + VP + Adjectival particle-paṭikkattakka (paṭi + kk + a + (t) + takka) ‘readable/suitable for reading’

Verb + (Tense) + Verbal Noun suffix

- Verb + Verbal Noun suffix - paṭittal (paṭi + ttal) ‘reading’
Verb + Tense + Verbal Noun suffix - *paṭittamai*

(*paṭi + tt + amai*) ‘reading’

**Adjective Wordform**

<table>
<thead>
<tr>
<th>Adjective + NG</th>
</tr>
</thead>
</table>

* e.g.

* Adjective + NG - *nallavaṇ* (*nalla + vaṇ*) ‘good – he’

**6.1.3.4. Sandhi rules involved in morphological parsing**

During the parsing of Tamil wordforms, the sandhi phenomena (both internal and external) have to be considered. Between root word and suffix or between suffix and suffix, the sandhi rules make an impact on the structure of the wordforms. Due to this sandhi, some phonemes may be added or deleted or changed.

For example, in the wordform *paṭittuppārttāṉ* ‘He tried reading’, there is a phoneme ‘p’ in between *paṭitu* (the verbal participle form of *paṭi*) and *pārttāṉ* (aspectual marker). This is due to the sandhi rule. When a root or suffix ends in *-PPu* and the next root or suffix begins with *PPu*, there should be a phonemic increment *p*. This is an addition of phoneme. During parsing of the above word, this phonemic increment should be identified and ignored. That is, only *paṭittu* and *pārttāṉ* should be considered for parsing purpose.

Here, the problem is, just by seeing two *ps* in a wordform, it could not be identified that one *p* is an increment due to sandhi. It is because, some wordforms may have two *ps* as part of the morpheme.
As like the addition or insertion of phonemes, there are deletions of phoneme or changes of phoneme in this Sandhi process.

e.g. *maram* ‘tree’ + *vēr* ‘root’ = *maravēr* ‘root of the tree’

- deletion of *m* in the first word.

*maram* ‘tree’ + *kal* (plural suffix) = *maraṅkal* ‘trees’

- change of the phoneme *m* into *ṅ*.

Before parsing the above wordform *maravēr*, the Sandhi deletion should be identified and *mara* should be reconstructed into *maram*.

Likewise before parsing the wordform *maraṅkal*, the *maraṅ* should be reconstructed into *maram*.

With some words, both deletion as well as addition of phonemes might have occurred due to sandhi.

e.g. *maram* ‘tree’ + *kiḷai* ‘branch’ = *marakkilai* ‘the branch of the tree’

Here, first the phoneme *m* of *maram* is deleted under a sandhi rule and then the phoneme *k* is added under some other sandhi rule. So, before parsing this wordform, these sandhi changes should be identified to get the original root *maram*.

Likewise, if the first word or suffix ends in a vowel and the following word or suffix begins with a vowel, then there will be an addition of phoneme *y* or *ṽ*. 

209
This is due to the rules of the syllabic structure.

e.g.  \( \text{teruvā ‘is it a street?’} = \text{teru ‘street’} + v + ā \) (interrogative clitic)

\( \text{ilaiyā ‘is it a leaf?’} = \text{ilai ‘leaf’} + y + ā \) (interrogative suffix)

Before parsing the above wordforms, these increments of \( y \) and \( v \) should be identified and the wordforms should be reconstructed as:

\[ \text{teru} + ā \text{ and ilai} + a \]

Sometimes, there may be some ambiguities.

e.g.  \( \text{ivvilai} \)

This could be parsed in two ways:

1.  \( \text{iv ‘this’} + \text{vilai ‘price’} \)
2.  \( \text{iv ‘this’} + \text{ilai ‘leaf’} \)

In (1), the second \( v \) is part of the word \( \text{vilai} \).

In (2), the second \( v \) is not part of the word, but only a glide occurred due to Sandhi rule.

In Tamil, with some wordforms, at the juncture of root word and the following suffix or between suffix and suffix, there may be some ‘fillers’ (“empty morph”).

e.g.  \( \text{maram ‘tree’} + \text{ai ‘accusative case suffix’} = \text{marattai ‘tree (acc.)’} \)

Here, in between \( \text{maram} \) and \( \text{ai} \), there is an empty morph \text{attu} (after the deletion of \( \text{am} \) from \( \text{maram} \). The sandhi rule is, the preceding word should be a noun ending in \( \text{am} \) and the following suffix should be a case one.
If the following suffix is not a case suffix, then the empty morph *attu* won’t occur.

*maram* ‘tree’ + ā (interrogative suffix) = *maramā* ‘is it a tree?’

All the above sandhi processes are called Internal Sandhi. There are also external sandhi processes in Tamil. This sandhi is called external sandhi.

In external sandhi, not only phonology, but morphology, syntax and even semantics are involved in Tamil.

e.g. *avay ennai* pārttup pēciṇāṇ ‘He saw me and talked.’

Here, in the second word *ennai* ‘me’ is added with a phoneme *p* at the end. The sandhi rule behind this, is: If a word is inflected for accusative case and the following word begins with a stop phoneme, then there should be an addition of a respective stop phoneme with the first word *ennai*.

And in the third word *pārttup* the final phoneme *p* is the result of the sandhi process. The rule is: if a word is in the *ceytu* pattern verbal participle form ending with two stop phonemes and the following word begins with a stop phoneme, then in the preceding word, there would be an addition of a stop phoneme – here, it is *p*.

Before parsing the above wordforms, the phonemes added because of the sandhi processes in the second word and in the third word, should be identified and deleted.

In some Sandhi processes, syntax is also involved.

1. *avay vēlai* pārttāṇ ‘He worked.’
2. *avay vēlaip* pārttāṇ ‘He saw the spear.’
In (1), the relation between the second word \( \text{vēlai} \) and the third word \( \text{pārttāṉ} \) is a non-casal one. So, there is no sandhi addition in the second word.

In (2), the relation between the second word \( \text{vēlai} \) and the third word \( \text{pārttāṉ} \) is a casal one. That is the segment \( ai \) is the accusative case suffix. So, there is a sandhi addition \( p \) in the second word.

The above sandhi process helps to consider the wordform \( \text{vēlai} \) in (1) as a root word, whereas the wordform \( \text{vēlai} \) should be parsed into \( \text{vēl} + ai \). But during parsing, the sandhi increment \( p \) should be deleted.

Thus, before parsing a wordform, if there is any sandhi change – addition, deletion, change of phonemes – or glide insertion or addition of some empty morph, they should be identified and the root wordform should be reconstructed for further parsing process.

6.1.3.5. Issues in Parsing

In parsing Tamil wordforms, there are some issues leading to ambiguities.

A single wordform may have more than one parser output. For example, in parsing the wordform \( \text{paṭittavarai} \), there would be two outputs.

1. \( \text{paṭittavar} \) ‘the learned person’ + \( ai \) (accusative case suffix)

2. \( \text{paṭitta} \) - the RP form of \( \text{paṭi} \) ‘study’ + \( \text{varai} \) ‘upto’

In (1), the inner parsing of \( \text{paṭittavar} \) is:

‘\( \text{paṭi} \) ‘study’ + \( tt \) (Past tense) + \( a \) (Relative Participle marker) + \( \text{var} \) (NG)’
So, the meaning of *paṭittavarai* is ‘the learned person (acc.)’

In (2), the inner parsing of *paṭitta* is:

‘*paṭi* ‘study’ + *tt* (Past tense) + *a* (Relative Participle marker)’

So, the meaning of *paṭittavarai* is ‘upto the studies done’

The aim of this present morphological parser is to find out all the possibilities of inflection found in a wordform. If a wordform gives place for more than one output, then the present parser should exhaust all these possibilities. In the above example, the root word *paṭi* has undergone two kinds of inflection. So, it should not stop its parsing process once it gets one structure. It should find out all the possible morphological structures found in the given wordform.

However, the morphological parser developed for spellchecking purpose could stop once it gets one possible structure. But, the parser developed for the present work – to translate the inflectional properties found in wordforms – should be different from the spellchecker parser.

Some more examples with more than one morphological structures:

**kaṭalai**
- *kaṭal* ‘sea’ + *ai* (acc.suffix)
- *kaṭalai* ‘nut’

**paṭikkavā**
- *paṭi* ‘study’+ *kk* (filler) + *a* (infī. suffix)+ *vā* (modal)
- *paṭi* ‘study’ + *kk* (filler) + *a* (infinitive suffix) + *v* (glide) + (clitic)

**neytāṉ**
- *ney* ‘ghee’ Noun + *tāṉ* (clitic)
- *ney* ‘weave’ Verb + *t* (Past Tense ) + *āṉ* (PNG )
6.1.4. Wordclass Tagger

The Parser output would be sent to another sub-module for wordclass tagging. Based on the root word and suffixes, it would decide the tagger of the wordforms. Mostly the final suffix of a wordform plays an important role in tagging.

The tagging of the wordform is important to decide the inflectional categories to which a particular suffix may belong. For example, the output of the parsing of the wordform ṭā would be:

1. ṭu ‘run’ + ā - negative relative participle suffix
2. ṭu ‘tile’ + ā ‘interrogative clitic’

In (1), ṭu is a verb root. Hence it could be followed by the negative relative participle suffix ā. As a verb root, it cannot be followed by the interrogative suffix. And the wordclass tagger of this wordform is Negative Relative Participle.

In (2), ṭu is a noun root. Hence it could be followed by the interrogative clitic ā. As a noun root, it cannot be followed by the negative relative participle. And the wordclass tagger of this wordform is Noun Interrogative.

Since the present work is concerned with the inflectional properties or suffixes, though the parser outputs are two (Negative Relative Participle and Interrogative noun), only the first output would be considered for further process.

Moreover, to decide the status and meaning of the individual suffixes inside a wordform, the wordclass category of the whole wordform is very much needed.
And the aim of the present work is to translate or find equivalences for the Tamil inflectional suffixes, the whole word category has an important role.

6.1.5. Variable Vs Invariable wordform

Based on the input provided by both the Parser and Wordclass Tagger, the next sub-module would decide whether the input wordform is a variable one or not.

If it is an invariable one, it would be discarded for further analysis. Only the variable wordforms would be considered for the present work.

For example, the following wordforms are invariable:

- mika ‘much’
- mikavum ‘much’
- oru ‘one’
- ayyō ‘Alas’

All the above belong to īṭaiccol (middle words). They won’t undergo any inflection process.

6.1.6. Inflection

The output of the previous sub-module – the filtered wordforms which are variable – are the input to this module. This sub-module would analyse whether the input wordforms are inflected or non-inflected.
For example, the following wordforms, though variables, are uninflected:

1. vēkamāka  ‘fast’
2. metuvākattāṅ  ‘slowly’
3. aḷakāṇa  ‘beautiful’

The root of (1) is a noun and it is attached with the adverb suffix āka.

The root of (2) is an adverb and it takes the emphatic clitic tāṅ.

The root of (3) is a noun and the attached suffix is the adjective suffix āṇa.

In all the above, the roots are variable since their wordforms are changed because of the attachment of suffixes. But these suffixes are not inflectional suffixes; they are only derivational suffixes and clitic.

On the other hand, the following wordforms are variable and inflected for various inflectional properties.

1. paṭittāṅ  ‘(he) studied’
    = paṭi + tt + āṅ
2. avaṇukku  ‘to him’
    = avaṇ + ukku
3. vantu  ‘having come’
    = vā + ntu
4. koṭutta  ‘one who gave’
    = koṭu + tt + a

The root of (1) is a verb and it is inflected for past tense and PNG.

The root of (2) is a pronoun and it is inflected for dative case.

The root of (3) is a verb and it is inflected for verbal participle.

The root of (4) is a verb and it is inflected for past tense and relative participle.

6.1.7. Inflected wordforms – with or without clitics

The output from the above sub-module is the input to this sub-module. That is, the variable as well as inflected wordforms are the input to this sub-module. Now, this sub-module would analyse these
wordforms to find out whether there are any clitics. If a wordform has clitics, it would be sent for filtering.

The following wordforms have variable lexemes as well as inflected ones. But there are some clitics in the wordforms.

1. \textit{avarkalukkuttan} ‘for them only’ = avar + kal + ukku + tāŋ
2. \textit{paṭittāŋa} ‘did he study?’ = paṭi + tt + āŋ + ā
3. \textit{paṭikkattāŋ} ‘to study only’ = paṭi + kk + a + tāŋ
4. \textit{avaṅukkumaṭṭum} ‘for him only’ = avaŋ + ukku + maṭṭum

In (1), in addition to the two inflectional suffixes, there is one clitic tāŋ.
In (2), in addition to the two inflectional suffixes, there is one clitic ā.
In (3), in addition to the two inflectional suffixes, there is one clitic tāŋ.
In (4), in addition to the inflectional suffixes, there is one clitic maṭṭum.

The clitics attached wordforms would be sent for filtration and the output without any clitic would be sent to next sub-module for further process.

6.1.8. Pure stem?

The input to the sub-module would be wordforms having both basic lexicons as well as derived ones as stems for further inflection. Basic lexicons mean here the root words available in Lexicon. The derived ones mean the stem having a root lexicon with some derived suffixes.

Since these derived stems are not available in the Lexicons, they would be sent to further process to get the meaning. Only after that they would be sent to the final Transfer module. If a wordform has only basic lexicon which is available in the Lexicon and inflected for some inflectional properties, directly it would be sent to the Transfer module.
In (1), the root noun talaivar is verbalized by the addition of the verbalizer ākku. Then the verb talaivarākku is inflected for tense īṉ and PNG ār. The verb talaivarākku is not a basic lexicon available in the Lexicon. It is a derived verb from the noun talaivar ‘leader’. So, to get the meaning of this derived verb, it should be processed first. Only after that, it could be sent to final transfer module.

In (2), the root word is pāl ‘milk’. It is added with the derivational suffix kāraṉ to get the derived word pālkkāraṉ. Since this word is not a basic lexicon, it may not be available in the Lexicon. To get the meaning of this derived word, it should be sent for further process. Here, ai is the inflectional suffix.

In (3), the root noun āḻaku ‘beauty’. It is added with the adjectival suffix āṉa to get the derived adjective āḻakāṉa ‘beautiful’. Then it is inflected for NG vaḷ and the resultant wordform is āḻakāṉavaḷ.

In (4), the root paṇpu ‘virtue’ is added with the derivational suffix uḷḷa to get the derived word paṇpuḷḷavar. It is further inflected for NG var and accusative case ai.
6.1.9. **Transfer Module**: This is the core module in the present work.

**Transfer Rule structure**

All the inflectional suffixes found in the input would be converted into abstract inflectional categories with the respective inflectional properties.

For example, *kaḷ* Plural → Plural

*tt* Past Tense → Past Tense

*iru* Perfect Aspectual → Perfect Aspectual

However, the root/stem would be retained with its category.

e.g. *paiyaṅ* Noun

*paṭi* Verb

*aḻakāṇa* Adjective

**Input**: Input for this module is the inflected wordforms. The stems of these wordforms may be either pure roots (‘lexicons’) or derived ones. And these wordforms would be free from any clitic since already they are, if any, filtered by some process earlier.

The stems are inflected for some inflectional properties. The input for this module would be:

**Pure root/derived stem + inflectional suffixes**

This module would process the input as follows:

It would get the English lexicon/root word for the Tamil stem, if it is a root lexicon, from the Tamil – English Lexicon. If the stem is a
derived one, by the previous sub-module, it would get the derived meaning and find the equivalent root word from the Lexicon.

Example 1:
Same number of inflectional suffix and same morphotactics:
paiyakoḷ

**Input:** paiyā + kaḷ (plural suffix)
paiyan is available in the Tamil – English Lexicon.

kaḷ - the plural suffix in Tamil has the equivalence ‘s’ in English.

**Input:** paiyā + Plural

The order of the noun root plus plural suffix in English is the same as in Tamil.

So, the output would be:

‘student’ – ‘s’

**Final output:** ‘students’

Example 2:
Difference in the number of suffixes but same morphotactics.
e.g. paṭittāṇ

**Input:** paṭi (Verb) + tt (pt.) + āy (3rd Person, singular, masculine suffix)

Here, paṭi is the root lexicon. When it is referred in the Lexicon, there are two categories – one is, noun paṭi; the other is verb paṭi.

However, since the root paṭi in the above example is followed by the tense suffix, it should be a verb, not a noun.

Now, even after deciding the root as a verb paṭi, there is a problem. There are two paṭi verbs in the Lexicon.
\textit{paṭi} (1) ‘be covered with’, ‘settle’

\textit{paṭi} (2) ‘read’, ‘study’

However, the first \textit{paṭi} belongs to the fourth group in the conjugation types.

\textit{kīr} - nt - v

The second \textit{paṭi} belongs to the eleventh group in the conjugation types.

\textit{kkir} - tt - pp

So, it could be decided that the verb \textit{paṭi} in the above input example should be the second one.

Now, the Tamil \textit{paṭi} in the example would be replaced by English verb.

\textit{paṭi} (Verb) + \textit{tt} (pt.) + ān (3\textsuperscript{rd} Person, singular, masculine suffix)

| ‘study’ (Verb) + past tense + 3\textsuperscript{rd} Person, singular, masculine suffix. |

The next step is, to find out the past tense inflectional suffix for the verb ‘study’. It is ‘ed’.

| ‘study’ (Verb) + ‘ed’ + 3\textsuperscript{rd} Person, singular, masculine suffix. |

Since in the past tense, there is no PNG marker in English, it would not be filled up. Anyhow that slot could be filled up by ‘(he)’ to indicate the Person, Number and Gender of the ‘Subject’.

| ‘study’ (Verb) + ‘ed’ + ‘(he)’ |

According to the spelling rule of English, the combination of the verb ‘study’ and the past tense ‘ed’ would give the word ‘studied’.

\textbf{Output:} ‘studied (he)’
Example 3:

Difference in the number of suffixes as well as in morphotactics:

\[ \text{paṭittirukkíṟāṇ} \]

| Input: | \( \text{paṭi} \) (V) + \( tt \) (VP-Simple) + \( īru \) (Perfect) + \( kkiṟ \) (Pr.) + \( āṇ \) (3rd PNG) |

In the above input, there are one Verb root plus four inflectional suffixes.

As like in the previous example, here also the English equivalence for the verb root \( \text{paṭi} \) is ‘study’.

\[
\text{‘study’} + \text{tt (VP-Simple suffix)} + \text{Perfect marker} + \text{Present tense suffix} + \text{3rd PNG}
\]

\[
\text{‘study’} + \text{Past Participle} + \text{‘have’} + \text{Present tense} + \text{‘3rd PNG.}
\]

\[
\text{‘studied’} + \text{‘has’} + \text{3rd PNG}
\]

Now, according to the word order of English, the aspectual word should precede the main verb. So, in the present example, the word order would be changed.

\[
\text{‘has’} + \text{‘studied’} + \text{3rd PNG}
\]

| Output: | ‘has studied-he’ |

In the present example, the present tense property has merged with the aspectual marker ‘have’ and the resultant form is ‘has’.

And since the aspectual ‘has’ occurred here, the main verb ‘study’ has changed into past participle form ‘studied’.

222
Example 4:
Modal Inflection:

\textit{varavēṇṭum}

\textbf{Input:} \texttt{vā ‘Verb’ + \texttt{a ‘infinitive marker’} + \texttt{vēṇṭum ‘obligatory modal’}}

\begin{itemize}
  \item ‘come’ Verb + ‘a’ infinitive marker + obligatory modal
  \item ‘come’ Verb + ‘a’ infinitive marker + ‘should’
  \item ‘come’ Verb + ‘should’
\end{itemize}

Now according to the word order rule in English, the auxiliary model verb should precede the main verb.

Also the main verb should be the first form of the verb.

In Tamil, according to the morphotactics, the main verb inflected for infinitive precedes the modal.

\begin{itemize}
  \item ‘should’ + ‘come’
\end{itemize}

\textbf{Output:} ‘should come’

Here, it is to be mentioned that most of the modals are not inflected for tense and PNG.

Example 5:

\textit{kollappaṭṭāṇ}

\textbf{Input:} \texttt{kol (main verb) + \texttt{a (infinitive suffix)} + \texttt{paṭu (passive marker)} + \texttt{t (past tense)} + \texttt{āṇ (PNG)}}
kill (main verb) + a (infinitive suffix) + auxiliary ‘be’ …past participle + past tense + āṉ (PNG)

‘kill’ (main verb) + a (infinitive suffix) + was …ed + āṉ (PNG)

‘killed’ (main verb) + was + ('he')

According to the word order rule, the auxiliary ‘was’ should precede the main verb.

‘was’ + ‘killed’ +('he')

Output: ‘was killed-he’

Example 6:

paṭittirukkavēṇṭum

input : paṭi (main verb) – ttu (VP-Simple suffix) – iru (Perfect Aspectual) – (kk)a (infinitive marker) - vēṇṭum (obligatory modal)

‘study’ (main verb) – ttu (VP simple) – have – (kk)a (infinitive marker) – should.

‘study’ (main verb) – en (Past Participle marker) – have – should

‘studied’ – have – should

According to the English word order, the modal should precede the aspectual and the aspectual should precede the main verb.

should – have – studied

Output : ‘should have studied’
Example 7:

paṭikkappaṭṭirukkavēṇṭum

**input:**

paṭi (main verb) – (kk)a (infinitive marker) - paṭu (passive marker) - ț (past tense suffix) - iru (Perfect aspectual) - (kk)a (infinitive marker) - vēṇṭum (model)

‘study’ (main verb) – (kk)a (infinitive marker) – be … en – have … en – (kk)a (infinitive marker) - should

According to the word order rules in English, the modal should precede the aspectual; aspectual should precede the voice; voice should precede the main verb.

should - have - been - studied

**Output:** ‘should have been studied’

Example 8:

paṭittu

**input:** paṭi (Main Verb) - ttu (VP simple marker)

study (main verb) – having ….ed

When a VP-Simple marker is followed by an aspectual within a wordform, it should be discarded during transfer to English, because the occurrence of the verbal participle form is dictated by the syntactic context, that is, by the following ‘Aspectual’ marker. The verbal participle marker does not contribute its inflectional property to the main verb here. But if it is the final of the wordform it should be retained because of its contribution of inflectional meaning to the verb, and translated into ‘having’.

According to the Word order rules in English, the auxiliary should precede the main verb.
having – studied

Output: having studied

Example 9:

paṭikka

Input: paṭi (Main Verb) - (kk) a (infinitive marker)

When the infinitive marker ‘a’ is followed by a modal auxiliary within a wordform, it should be discarded during translation into English, because the occurrence of the verbal participle form is dictated by the syntactic context, that is, by the following ‘Modal’ marker. The verbal participle marker does not contribute its inflectional property to the main verb here. But when it is the final in a wordform, it should be retained because of its contribution of inflectional meaning to the verb. It should be translated into English Infinitive, indicating word ‘to’.

‘s’study’ – to

Output: to study

Example 10:

paṭittāl

Input: paṭi (Main Verb) - ttu (VP Simple marker) – āl (conditional marker)

‘s’study’ (main verb) - … ed (past participle marker) – if (conditional word)

If - study - ed

Output: If studied
Example 11:

\( paṭittāl \)

**Input:** \( paṭi \) (main verb) + \( ttu \) (VP simple marker) + \( āl \) (cond. marker)

‘study’ (Main Verb) + \( ttu \) (VP simple marker) + \( āl \) (conditional marker)
‘study’ (Main Verb) + \( ttu \) (VP simple marker) + ‘if’ (conditional word)

Here, the translation \( ttu \) VP Simple marker depends upon the time or tense of the main clause.

e.g. \( avaṉ\ paṭittāl, ṇānum\ paṭippēn \) ‘If he studies, I will study.’
\( avaṉ\ paṭittāl, ṇānum\ paṭikkirēn \) ‘If he studies, I study.’

So, the translation should be ‘studies’ and ‘studied’

‘if studies’

**Output:** If studies

The relevant flow chart, flow diagram and some screen shots are provided here.

1. Tamil Noun Morphotactics
2. Tamil Verb Morphotactics
3. Flow Diagram of Transfer Module
4. Program Sample Outputs
Thus, the Transfer module could translate any Tamil inflected word into English word (e.g. *paiyānkal* – ‘boys’) or word with grammatical or functional word (e.g. *avaṇukku* – ‘to him’).

Once the Tamil input is given to this module, it would do everything (Tokenization, Morphological Parsing, POS Tagging etc.) to arrive at the final English translation output. If the input has any clitic or derivational suffixes, this module could handle them effectively to get the final inflected form to be translated to English.