*J. Indian Inst. Sci.*, Mar.–Apr. 1990, **70**, 145–158.


# Phrase markers in Indian languages

N. ALWAR AND S. RAMAN*
Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600 036, India.

Received on June 7, 1989; Revised on October 5, 1989.

**Abstract**

Analysis of a natural language input typically consists of the stages of syntactic and semantic analysis. The standard technique is to use a context-free grammar (CFG) and an augmented transition network (ATN), and build a parse tree before processing further. Such a parse tree consists of noun-phrases (NP) and verb-phrases (VP), which are centered around a noun or a verb accordingly. In this paper, we suggest an alternative approach to the concept of *phrases* which appears to be more meaningful in the context of Indian languages. Instead of the conventional NP-VP grammars, we adopt verb boundaries as *phrase markers* and build phrase-level representations to convey the meaning of a sentence. The validity of the approach is established through studies on the application-oriented experiments carried out for Indian languages.

**Key words:** Phrase, phrase-level semantics, phrase markers, frame-based representation, verb-centered.

## 1. Introduction

Translation is the process of transforming a sentence $S$ of language $L$ into a synonymous sentence $S'$ of language $L'$. One of the ways of doing this transformation is to use a context-free grammar (CFG) that defines the different syntactic categories of the words used in the languages $L$ and $L'$, and also the order in which these categories combine to form a sentence in these languages. In such cases, the structure of the source language is mapped on to the structure of the target language, but understanding of the sentence may not be attempted. But, we believe that the rules stated in such a grammar are not sufficient for translation. Though a grammar is necessary for the valid construct of the sentences, it presents some ill-fitting concepts that can hinder correct planning for translation. The arguments given below illustrate this concept.

Some grammars define a subject and an object semantically, and declare that the subject is that part of a sentence about which the rest of the sentence conveys something. Some other grammars assert that one can identify the subject of the sentence by asking questions

* Present address: Department of Electrical Engng and Computer Science, The George Washington University, Washington, D.C. 20052, USA.

like 'who' or 'what' followed by the verb in the sentence. Essentially, both these views try to extract the information content in a sentence although their starting points are different. This shows that it is more important to get at the information content. This can be reliably had only if the domain of discourse, the context in which the sentence is spoken, and the intentions of the speaker and the listener are known. This, in turn, implies that the meaning of the sentence should be derived. This can be accomplished by identifying the relation of the verb of a sentence with other components in the sentence, since the verb plays an important role in conveying the meaning of a sentence. Hence, rather than using a standard CFG, a verb-centered analysis is advantageous in understanding the meaning conveyed by a sentence. Further, the use of a CFG restricts the type of sentence and the syntactic ordering of the different words in a sentence. Since Indian languages have a relatively free-word order, which permits a sentence to be rephrased in many ways, it suggests that we can devise a new approach which gives more importance to the verb and which derives the meaning of the sentence using the relationship of other words with reference to the verb. In this paper, we discuss one such approach.

## 2. Background on verb-centered approaches

A number of verb-centered formalisms are already in use. The case grammar theory developed by Fillmore[1] and the conceptual-dependency scheme introduced by Schank[2] are the most popular verb-based approaches. The case grammar defines a number of case categories, and splits the input sentence into each of those categories. For example, consider a system having the following cases:

| | | | |
|---|---|---|---|
| Agent | Action | Dative | Locative |
| Time | Instrumental | Source | Goal |

An input sentence like,

*During yesterday's match Ram hit Shyam with his bat,*

will be analysed and the following case categories will be assigned.

| | | | |
|---|---|---|---|
| Agent | : *Ram* | Action | : *hit* |
| Dative | : *Shyam* | Instrument | : *Ram's bat* |
| Time | : *yesterday* | Location | : *playground* |

All the above cases, except 'Location' are derived from their order of occurrence and the prepositions. The conventional case-grammar-based system uses sufficient dictionary information to derive that matches are played at 'playground'. It may be noted in passing that domain-specific approaches will disambiguate among the several meanings of the word 'match'.

Splitting a sentence into such case categories eases further processing. The reasons attributed for this are that they are of language-free nature and that there is an explicit representation of the meaning.

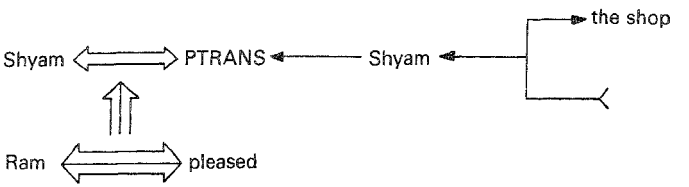


FIG. 1. A sample CD-based representation.

In the conceptual dependency (CD) approach, the basic unit is the 'conceptualization' which consists of an action and some links with the objects taking part in the action. The links have a specific set of meanings. Some of the objects involved in the action themselves can be sets of linked objects. The types of links and the built-in predicates make CD more suitable for representing the meaning of texts, which deal with everyday life, than for that of technical texts. A sample CD-based representation for the sentence

*Ram wanted Shyam to go to the shop*

is shown in fig. 1.

### 3. The concept of a phrase

In this background of verb-centered analysis, we propose a different approach for processing sentences in Indian languages. This approach was used as part of our studies on machine translation between Indian languages. It was an outcome of certain characteristics of Indian languages, and of our hypothesis based on them, as listed below.

1. Indian languages are highly case-inflected. The verbs as well as nouns undergo inflections in these languages. In many situations, the inflections themselves disambiguate the semantic content of the sentences.
   For example, the following sentence in English is syntactically and semantically ambiguous:
   I saw a man on the hill with the telescope.
   When this sentence is translated and is written in Tamil, the sequence of words and particularly their inflections disambiguate the semantics as illustrated below in their transliterated form. (The transliterated codes for Tamil and Hindi are given in Appendix I).

   *Ndaan taelascoppen muulam maliyen maael erundtha oru manethani paarththaen.*
   *Ndaan maliyen maael taelascopputan erundtha oru manethani paarththaen.*
   *Ndaan taelascoppu erundtha maliyen maael oru manethani paarththaen.*

   These sentences respectively indicate that the telescope was used to spot a man, or that the man had a telescope, or that the telescope was installed on a hill.

2. The feature of the relatively free-word order permits a sentence in an Indian language to be rephrased in many ways, while still conveying the required meaning.
   For example, the sentence in Tamil,

   *Ndaan ndaalzi kaali tellekku poovaaen.*
   can be rephrased in the following ways.
   *Ndaan tellekku ndaalzi kaali poovaaen.*
   *Ndaalzi kaali ndaan tellekku poovaaen.*
   *Ndaalzi kaali tellekku ndaan poovaaen.*
   *Tellekku ndaan ndaalzi kaali poovaaen.*
   *Tellekku ndaalzi kaali ndaan poovaaen.*

   This shows that although there is free-word ordering, between certain groups of words as in 'ndaalzi kaali' the word ordering cannot be changed.
3. The above example also shows that the verb always appears at or towards the end of the sentence. Although it is permissible to locate the verbs in different places in a sentence, such sentence constructions are not generally in vogue.

The above characteristics of Indian languages led us to think of a non-conventional approach. It resulted in our phrase-level approach which is based on the following hypotheses.

1. The fact that we understand ungrammatical sentences to a large extent shows that the relevance of a strict grammar can be underplayed in sentence understanding.
2. The fact that human interpreters do not wait for the completion of a full sentence, but instead start translating as soon as they get certain bits and pieces of information, shows that it may be sufficient to understand a sentence in parts.

In the light of these observations, we introduce the concept of *phrase-level semantics*. The term *phrase* as we refer to is different from the conventional verb/noun phrases. In our case, the phrases are marked by verb-boundaries. The portion of a sentence that is to the left of a verb, or between two verbs, is termed a *phrase*. Analysis of sentences is restricted to the phrases identified as stated above. The difference between the conventional concepts of NP-VP and the proposed concept of a phrase is illustrated through the following example.

A conventional parsing of the sentence,
*Ram climbed up the hill,*

results in a syntactic structure as shown in fig. 2.

This illustrates that there are multiple noun/verb phrases in a sentence, and the assignment of the semantic roles to these phrases is complicated (*i.e.*, relationships between phrases have to be explicitly stated). This complexity can be reduced if we can view this sentence as a single-phrase sentence. Adoption of such a view is possible by using our *phrase-level* formalism when we consider the Tamil equivalent of the sentence in the above example. This Tamil sentence reads as:

*Raam maliyen maael aaedrenaan.*

When this sentence is analysed based on our phrase-level approach, the whole sentence is

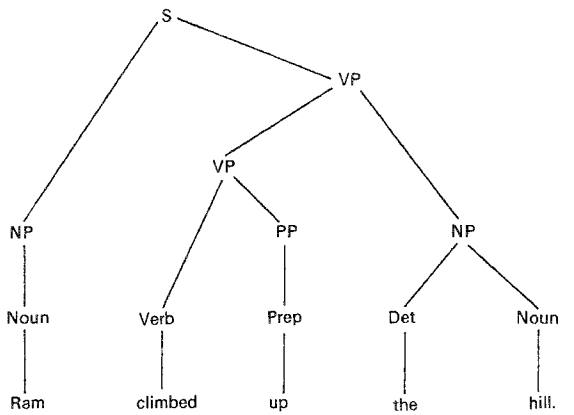


FIG. 2. A typical parse tree.

treated as a single phrase with the inflected verb 'aaedrenaan' as its delimiter. It results in a representation that directly conveys the meaning of the sentence. The frame structure produced thus is as shown below:

| | |
|---|---|
| Action | : Climb |
| Tense | : Simple past |
| Agent | : Raam |
| Gender | : Masculine |
| Number | : Singular |
| Dative | : Hill |

In tune with the philosophy that the representation of meaning is language-free, our system derives the entries as shown above in English itself, irrespective of the input language. The dictionary entries have been appropriately designed to aid this derivation. The contents of our dictionary entries are given in the details of implementation.

This example illustrates two aspects of the approach. Firstly, the analysis does not give importance to the syntax or, more specifically, the word ordering in the sentence. The case markers are given more importance, and the relationship between the verb and each of the other words is derived using the case grammar mainly. Secondly, it may be possible to make the system accept ill-formed inputs by relaxing the syntax-checking rules in the parser. This ability of the parser to accept ungrammatical inputs is owing to the expectation-driven nature of the parser. First, the verb is spotted and, with reference to it, other words are 'anticipated'[3].

A frame structure is available *apriori* for each of the verbs used in a specified context. The

parser analyses each phrase in a sentence with a view to fill the slots in this frame structure. The philosophy of restricting the analysis to verb-marked phrases is apt from the point of view of the verb-final structure of Indian languages. Further, the free word-ordering aspect of the Indian languages lends support to such an expectation-driven parsing. In case of simple sentences consisting of one verb, possibly accompanied by an auxiliary verb as a qualifier, the output of the parser directly represents the meaning of the sentence. However, in case of multiple-phrase sentences, the first phrase generally is the main part of the sentence, and the remaining phrases are either qualifiers or have independent frame structures depending on their complexity. For example, the sentence

*Ndaan katikku pooka vaaendtum*

is split into the following two phrases:

1) *Ndaan katikku pooka,*
2) *vaaendtum.*

The second phrase consists of only a verb which is an additional qualifier of the main verb, and so the resultant representation is a single frame structure as shown below.

| | |
|---|---|
| Action | : Go |
| Tense | : Simple future |
| Destination | : Shop |
| Agent | : I Person |
| Number | : Singular |
| Gender | : Masculine (by default) |
| Aux. verb | : Want |

Consider the following conjunctive sentence.

*Raaman katikku pooyvettu varum vazheyel*
*Seethaavetam oru puththakam vaangke vandthaan.*

It will be split into the following four phrases.

1) *Raaman katikku pooyvettu,*
2) *varum,*
3) *vazheyel Seethaavetam oru puththakam vaangke,*
4) *vandthaan.*

Here, the phrases 1 and 3 are similar to the sentence considered in the previous example. Phrase 2 is a qualifier of phrase 1, and phrase 4 is that of phrase 3. The individual representations of these phrases and the resultant representation of the whole sentence are shown in figs 3 and 4.

The final representation is the concatenation of the four individual phrase-level representations. The combination of the individual phrase meanings to form a sentence meaning is based on the concept of 'compositionality'. Compositionality refers to the fact that the meaning of a sentence is largely the combination of the meaning of each of the individual words. In our case, this concept has been extended to phrases.

| | | | |
|---|---|---|---|
| 1) Action | : Go | 2) Action | : Come |
| Tense | : Simple past | Tense | : Simple future |
| Destn | : Shop | | |
| Agent | : *Raaman* | | |
| 3) Action | : Get*Buy | 4) Action | : Come |
| Source | : *Seethaa* | Time | : Simple past |
| Object | : A book | Person | : III Person |
| Location | : Pathway | Gender | : Masculine |
| | | Number | : Singular |

FIG. 3. Phrase-level representation.

| | | | |
|---|---|---|---|
| Action | : Go | | |
| Tense | : Simple past | | |
| Agent | : *Raaman* | | |
| Destn | : Shop | | |
| Aux. verb | : Come | Action | : Get/Buy |
| | | Tense | : Simple past |
| | | Person | : III Person |
| | | Gender | : Masculine |
| | | Number | : Singular |
| | | Location | : Pathway |
| | | Source | : *Seethaa* |
| | | Object | : A book |
| | | Aux. verb | : Come |

FIG. 4. Combined representation.

The concept of our phrase-level semantics thus helps in analysing natural language input sentences consisting of multiple phrases. This technique may be extended so that multiple sentences may be processed in order to understand a passage or a paragraph.

## 4. Implementation and results

In order to validate the use of 'phrases' and 'phrase-level semantics' as discussed above, a specific domain of application was chosen. The domain chosen in our case was the typical conversation carried out at a railway counter. Typical sentences generated in that domain were subject to the analysis as outlined above. The system was implemented on IBM PC series, and the overall block schematic of the system is shown in fig. 5. Currently the system carries out Tamil to Hindi translation for the stated objective. The system consists of the following five stages.

1. Multilingual front-end
   This handles the complex character shapes of Indian languages, and allows the users to enter texts in Indian languages. The translated output is also displayed in the respective language, and thus provides the required user interface[4].
2. Coder
   The input Tamil text in a specific coded form is converted into transliterated codes which are needed for the search through the dictionaries.

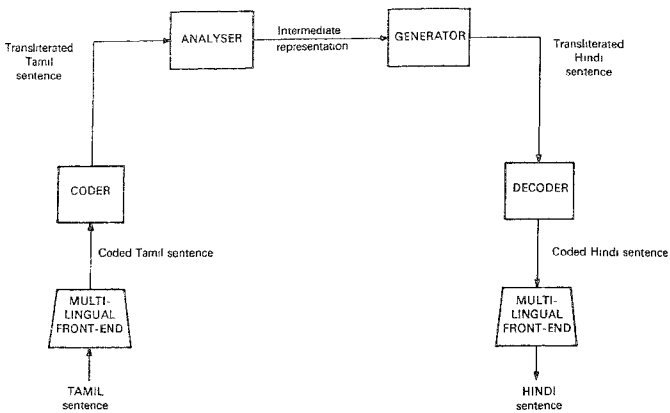


FIG. 5. Tamil-to-Hindi translation system—Block schematic.

3. Analyser
   This is the heart of the system which implements the phrase-level analysis for deriving the intermediate representation.

4. Generator
   The intermediate representation is used to generate the corresponding Hindi sentence which is displayed in the transliterated format.

5. Decoder
   The transliterated Hindi sentence is decoded for display of the Hindi sentence with the help of the multilingual front-end.

### 4.1 *Analyser*

The analyser uses a total of nine dictionaries containing the following word categories:

| | | |
|---|---|---|
| ·VERBS | ·NOUNS | ·PRONOUNS |
| ·QUERY | ·ADJECTIVES | ·ADVERBS |
| ·COUNT | ·PLACES | ·TIME |

All the dictionaries use the same structure and contain the root, meaning, *sandhi** and inflections. In the case of Tamil, *sandhi* denotes the tense informaion and the inflections contain information about the number and the gender. A typical dictionary entry for the verb *poo* (go) is shown below.

**Sandhi* (Sanskrit). Euphonic junction or coalition.

ROOT = *poo* MEANING = GO
*k* *um, a*
*n* *aaen, aan, aalz, aarkalz, eerkalz, athu, a, oom*
*v* *aaen, aan, aalz, aarkalz, eerkalz, athu, a, oom*
*kedr* *aaen, aan, aalz, aarkalz, eerkalz, athu, a, oom*
*kendr* *aaen, aan, aalz, aarkalz, eerkalz, athu, a, oom*
*y* —
ENDROOT

The morphological analyser uses the above entries for checking the valid constructions of the verbs/nouns. Thus a word may be rejected if it does not match with any of the above categories. However, this can be refined further for the acceptance of ill-formed sentences by allowing the analyser to accept a default *sandhi* and inflections, in case of any mismatch. Currently, this feature has not been implemented.

The analyser selects a frame structure for each of the verbs[5]. These frame structures are predefined and form an integral part of the program. The frame contains the relevant case slots for that verb. A typical frame structure for the verb *poo* (go) is shown in Table I.

Each of the slots may contain subslots. For instance, the slot 'time' carries day, data and hours as subslots. The expectation-driven analyser uses this frame structure and analyses the individual phrases with a view to fill these slots for each phrase.

The stage-by-stage analysis of an input sentence is explained for the following sentence.

*Ndeengkalz aendtha traeyenel pooka verumpukereerkalz?*
(By which train do you wish to go?)

The first step in the analysis is to spot the main verb. Hence the analyser searches for the presence of a verb in a given input sentence. In the above sentence, it encounters the root word *poo* (go) of the inflected verb *pooka* (to go). The morphological analyser decodes this word and returns the root, *sandhi* and the inflectional forms together with the meaning of the root word. The analyser then generates a frame structure containing predefined slots. The choice of this frame is based on the meaning of the root word. These slots hold the

**Table I**
**Frame structure for the verb *poo* (go)**

| Slot | Description |
|---|---|
| Mood | Mood of the sentence (Statement, query, etc.) |
| Agent | Performer of action |
| Mode | Mode of travel |
| Source | Starting place |
| Destination | Destination place |
| Time | Time at which action takes place |

possible information along with this root. The position of the main verb delimits a phrase and hence the sentence is split into the following two phrases.

1) *Ndeengkalz aendtha traeyenel pooka*
2) *verumpukereerkalz*?

The analyser then concentrates on the first phrase. The slots in the frame are formed as a prediction list. The first entry in this list is the query. A sentence is classified as a query if it contains a query word and if it ends with a question mark (?). A query search in the prediction list is honoured only if the sentence had ended with a question mark. As in this example, a search for a match with a query word is performed. The query dictionary which contains all the query-related words is used for the search, and this returns a success on encountering the word *aendtha* (which). The spotting of this query creates another prediction list that is generated by the built-in grammar rule base. This rule indicates that the word next to *aendtha* must be a noun which is classified as the query object. In this example, it matches with the word *traeyenel*. A case analysis of this noun deciphers this word as 'TRAIN' and the relation of this noun with the query as 'IN'. The next item in the prediction list is the agent. The search in the pronouns dictionary identifies the agent. This builds the semantic representation gradually and ends up in a phrase-level representation.

The second phrase of the sentence contains only the auxiliary verb that describes the mood of the sentence. Hence, it is appended to the representation without any special link to the first phrase. The associated number, gender and tense categories are checked for validity and added in the final representation, as shown in fig. 6. The sample set of sentences handled and their representations are given in Appendix II.

### 4.2 *Generator*

The output of the analyser forms the input to the generator. Since the intermediate representation is language-free, this can be used for translation of the source language into any target language. In our case, it was translated into Hindi. Rather than applying the target language grammar rules directly on the intermediate representation, it is first translated into a target language representation and then synthesized to form the Hindi sentence.

The generator first translates only the root word in the intermediate representation into the target language using an intermediate language (IL)–target language (TL) dictionary.

```
Sentence type    : Query
Action           : Go
Tense              : Simple future tense
Agent            : II Person
  Number           : Plural/Singular respect clause
Query type       : Which
  Query object     : Train
  Relation         . In
Aux. verb        . Desire
```

FIG. 6. Final representation of the sentence: '*Ndeengkalz aendtha traeyenel pooka verumpukereerkalz?*'

This results in the TL-representation. It is then synthesized using a multidimensional dictionary that stores the inflections to be used for different tense, number and gender. The inflected words are concatenated using a fixed word-order strategy to form the translated sentence[6]. The free-word-ordering feature of the Indian languages allows us to implement such a simple form of generator.

### 4.3 *Results*

The above system was initially tested for a set of 12 Tamil sentences. The sentences and the dictionaries were created using the transliterated codes. The initial results were so highly successful that they validated our phrase-level approach. The analyser's ability was tested further by letting it handle more types of sentences. During this extension stage, the system had to be refined so that it may handle the different case categories. The system presently accepts 64 types of sentences.

A similar step-by-step refinement was carried out for the generator also. The generator produced the translated sentence in transliterated form. A multilingual front-end and a coding/decoding module were added to the system. The Tamil-to-Hindi sample translations are also given in Appendix II.

## 5. Conclusion

In this paper, we have outlined the conventional approach of identifying the phrases in a sentence, and developed our concept of phrases which can be more meaningful in the processing of Indian languages. Our phrase-level approach is based on the concept that the verbs convey the action part in a sentence, and that compositionally other components of a sentence find their respective places. Thus a sentence is considered as a collection of phrases, each of which is delimited by a verb. The validity of this approach is established through studies on application-oriented experiments carried out with specific reference to translation from Tamil to Hindi. The extension of 'phrase-level semantics' to passage understanding is identified as the scope for future work.

## References


1. FILLMORE, C. J. — The case for case, in *Universals in linguistic theory*, (eds) E. Bach and R. Harms, 1968, Holt Rinehart and Winston, New York.

2. SCHANK, R. C. — *Conceptual information processing*, 1975, North-Holland.

3. RAMAN, S. AND ALWAR, N. — Studies on phrase-level semantics as applied to machine translation in Indian languages, (eds T. O'Shea and V. Sgurev), in *Proc. Intl Conf. on Artificial Intelligence-Methodology, System and Applications (AIMSA 88)*, Bulgaria, Sept. 1988, pp. 313–318.

4. ALWAR, N. *et al* — A multipurpose multilingual package for Indian languages, *Proc. of the Regional Workshop on Computer Processing of Asian Languages*, Bangkok, Sept. 1989, pp. 18–26.

5. RAMAN, S. AND ALWAR, N. An AI-based approach to machine translation in Indian languages, paper accepted for publication in *Commun. ACM*, May 1990.

6. ALWAR, N., RAMAN, S. AND DAMOR, B. R. A natural language generator for Hindi, *Proc. of the Regional Workshop on Computer Processing of Asian Languages*, Bangkok, Sept. 1989, pp. 102–109.

## Appendix I

### Transliteration codes for Tamil

| அ | ஆ | இ | ஈ | உ | ஊ | எ | ஏ | ஐ | ஒ | ஓ | ஔ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| a | aa | e | ee | u | uu | ae | aae | i | o | oo | au |

| க் | ச் | ட் | த் | ப் | ற் | ங் | ஞ் | ண் | ந் | ம் | ன் |
|---|---|---|---|---|---|---|---|---|---|---|---|
| k | c | t | th | p | dr | ng | ghy | dn | nd | m | n |
| ய் | ர் | ல் | வ் | ழ் | ள் | ஸ் | ஜ் | ஷ் | க்ஷ் | ஜ் | ஸ்ரீ |
| y | r | l | v | zh | lz | s | h | sh | ksh | j | sree |

*e.g.* கி = க் + இ = k + e ke

ஏறி = ஏ + ற் + இ = aae + dr + e = aaedre

### Transliteration codes for Hindi

| अ | आ | इ | ई | उ | ऊ | ए | ऐ | ओ | औ |
|---|---|---|---|---|---|---|---|---|---|
| a | aa | e | ee | u | uu | ae | i | o | au |

| क | ख | ग | घ | ङ | च | छ | ज | झ | ञ |
|---|---|---|---|---|---|---|---|---|---|
| k | kh | g | gh | ng | c | ch | j | jh | ghy |
| ट | ठ | ड | ढ | ण | त | थ | द | ध | न |
| t | dt | d | dh | dn | th | dth | td | tdh | n |
| प | फ | ब | भ | म | य | र | ल | व | स | ह | ष | क्ष |
| p | ph | b | bh | m | y | r | l | v | s | h | sh | ksh |

*e.g.* कि = क + इ = k + e = ke

जाना = ज + आ + न + आ = j + aa + n + aa = jaanaa

**Appendix II**

Sample set of Tamil sentences, their representations and the Hindi translations

1. Tamil sentence : *ndaan ndaalzi tellekku poovaaen.*
   Intermediate representation

| | |
|---|---|
| Main verb | : GO |
| Tense | : Simple future tense |
| Sentence type | : STATEMENT AFFIRMATIVE |
| Agent | : I Person |
| Number | : Singular |
| Details of time | : Tomorrow |
| Destination | : Delhi |

Hindi translation : *mi kal dellee jaauungaa.*

2. Tamil sentence: *ndaangkalz ndaaedrdru koovi aeksperassel caaelathukku poonoom.*
   Intermediate representation

| | |
|---|---|
| Main verb | : GO |
| Tense | : Simple past tense |
| Sentence type | : STATEMENT AFFIRMATIVE |
| Agent | : I Person |
| Number | : Plural |
| Details of time | : Yesterday |
| Mode of transport | : Koovi Express |
| Relation | : BY/IN |
| Destination place | : Salem |

Hindi translation : *ham kal koovi aekspraes sae saelam hayae dthaae.*

3. Tamil sentence : *ndeengkalz tellekku aeppozhuthu pooka vaaedntum?*
   Intermediate representation

| | |
|---|---|
| Main verb | : GO |
| Tense | : Simple past tense |
| Sentence type | : QUERY |
| Query type | : When |
| Agent | : II Person |
| Number | : Plural |
| Destination place | : Delhi |
| Auxiliary verb | : Want |

Hindi translation : *aap kab dellee jaanaa caahthae hin?*

4. Tamil sentence : *ndaan mattumaavathu tellekku pooka muteyumaa?*
   Intermediate representation

| | |
|---|---|
| Main verb | : GO |
| Tense | : Simple future tense |

| | |
|---|---|
| Sentence type | : QUERY |
| Query type | : YES/NO |
| Topic | : At least |
| Agent | : I Person |
| Number | : Singular |
| Destination place | : Delhi |

Hindi translation: *mi kyaa kam sae kam dellee jaauungaa?*

5. Tamil sentence: *endru avvalzavu kuuttam elli.*
Intermediate representation

| | |
|---|---|
| Main verb | : BE |
| | NEGATED form of verb |
| Sentence type | : STATEMENT NEGATION |
| Topic | : So much |
| Details of time | : Today |
| Agent | : Rush |

Hindi translation : *bheed aaj bahuut nahe hi.*

6. Tamil sentence: *ungkalzukku aeththani tekkaet vaaentum?*
Intermediate representation

| | |
|---|---|
| Main verb | : WANT |
| Tense | : Simple future tense |
| Sentence type | : QUERY |
| Query type | : How many |
| Query object | : Ticket |
| Agent | : II Person |
| Number | : Plural |

Hindi translation: *aap koo kethnae tekat caaheyae?*