Chapter 4

Mapping Dependency Structures

4.1 Introduction

A sentence can be analyzed at different levels. Parsing is one way to analyse a sentence at the level of syntax or semantics. A constituency parse analyses sentences at the level of syntax. It assigns a structure to the words in a sentence in terms of syntactic units. As the name suggests, a constituency parse organizes words into nested constituents i.e. it divides the words of a sentence into sub-units called phrases. Whereas, a dependency parse analyses sentences at the level of semantics. Dependency structure represents words in a sentence in a head modifier structure, and also attests relation labels to the structures.

Dependency parse is preferred over constituency parse particularly for morphologically richer languages. It is more suitable for a wide range of Natural Language Processing (NLP) tasks such as machine translation, information extraction, question answering, etc. [85, 48, 46]. Also, it is very easy even for a non-linguist to think in terms of relation extraction whereas the constituency parse representation seems much more foreign and forbidding [48]. Parsing dependency trees is simple and fast [115].

In order to adapt a particular dependency modal into some other modal, the need of some type of modifications has been always realized. For instance, Venkatapathy (2010) in his thesis stats the need of modifying Stanford Dependency graph output to use it in his syntax based model. He had to perform various operations like removing cycles, grouping auxiliaries and determiners, adding punctuations and conjuncts to the dependency structure etc. on Stanford dependency parse.

In this chapter, we will present the need of the Anusāraka dependency schema that brings in the output of various English parsers into one uniform notation. The Anusāraka dependency schema is based on ingenious concepts from the Pāṇinian Grammar (PG) which offers a sound theory to identify the grammatical relationships among the words in a sentence for linguistic analysis. The main importance of PG is in observing the information coding and its flow in language. Apart from discussing the notion of *abhihita* (expression of a grammatical relations) from English point of view, we will take up some of the concepts such as root/stem (*prakṛti*), affixes (*pratyaya*), *sup, tiṅ, pada*, and *samasta-pada* from

Chapter 3, and examine how these concepts can help design simple and contentful dependencies. We will examine the Anusāraka dependency representation and its underlying design principles.

The main objective behind this study is to develop a uniform schema that can map any English dependency parser's dependency labels into Anusāraka dependency schema with some modifications. Bringing any English dependency parser labels into Anusāraka dependency schema has two fold objectives:

- A uniform representation for any English parser
- Develop a simplified approach for word ordering and word generation rules for English to Indian language MT systems

It is known that different parsers have different strengths [94]. Since they are based on different approaches, they have different degrees of output quality for different syntactic constructions. One can obtain maximum benefit if the strengths of all the parsers can be combined. Thus, if someone wants to combine the strengths of all the parsers, one needs a common representation.

4.2 English Parsers

Several dependency parsers such as Stanford Parser, Link Grammar Parser (LGP), CLEAR Dependency Parser, Minipar, MSTParser, Malt and many more are available for English. However, no two dependency parse output formats match with each other. These parsers have no agreement on the names of the dependency labels, their representation style and the number of dependency labels [17]. That is why, they have no uniform dependency tree structure as well. In this section, we will give a very brief introduction to some of the dependency parsers and show their dependency representation styles using sentence 29.

(29) She is staying in the hostel.

- **Stanford Parser:** It is a statistical parser. It contains a set of around 50 dependency labels [47]. It provides directed arcs among the words which form a tree. To some extent, it attempts to give the relationships between content words. It gives five variants of the typed dependency representation which range from a more surface-oriented representation to a more semantically interpreted representation [47]. In order to mark the grammatical relationships, it tries to use the notions of traditional grammar so that the relations can be comprehended easily by the users [48]. See Figure 4.2 to get an idea about its dependencies:

  In 2014, Stanford parser adapted universal grammatical relations into its dependency representation [47]. Universal dependencies is a cross-linguistically consistent annotation schema for many languages [95]. Anotated treebanks for 33 languages are currently available under this tagging scheme. Stanford parser generates two types of dependency parses: (i) Basic Dependencies and

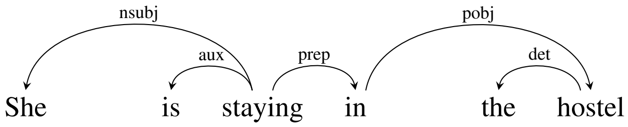


Figure 4.1: Stanford basic dependencies

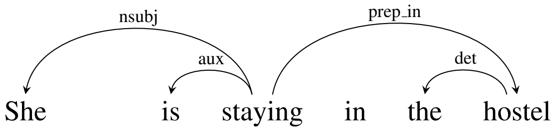


Figure 4.2: Stanford collapsed dependencies

(2) Enhanced Dependencies using Universal dependencies annotation. Figure 4.3 and 4.4 show Universal Stanford Dependencies parses for sentence 29

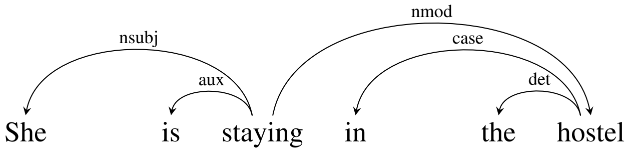


Figure 4.3: Basic Universal Stanford dependencies

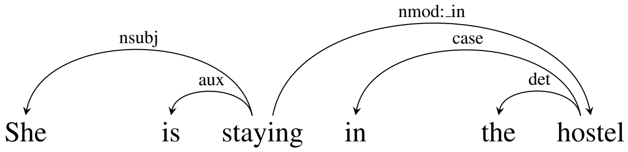


Figure 4.4: Enhanced Universal Stanford dependencies

- **CLEAR Parser:** It takes input as constituent-based trees in Penn Treebank style and converts them into dependency trees. It adapts the Stanford dependency approach as its core structure. It has borrowed few relation labels such as ‘hmod’ (modifier in hyphenation), ‘hyph’ (hyphen) and ‘nmod’ (modifier of nominal) from the CoNLL dependency approach. It also introduces some new labels like ‘intj’ (interjection) and ‘meta’ (meta modifier) to minimize unclassified dependencies [35, 36]. Figure 4.5 shows CLEAR dependencies for sentence 29.

  One notices that CLEAR dependencies match with the earlier version of basic Stanford dependencies for the sentence 29. But, as said before, there are labels which they have borrowed from CoNLL and they have introduced some new labels.

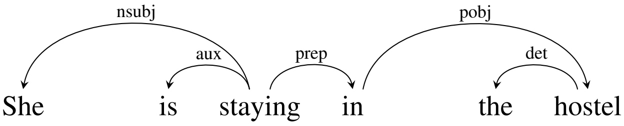
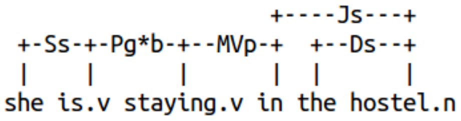


Figure 4.5: CLEAR dependencies

- **Link Grammar Parser (LGP):** It is a rule based parser. A link grammar consists of set of words, each of which has a linking requirement. The linking requirements of each word are contained in a dictionary, hence, constructing grammar for irregular verbs of English is easy. Its grammar contains roughly eight hundred formulas and 25,000 words to capture different phenomena of English grammar. It has around 106 labels called linkages or connectors. A connector name starts with one or more upper case letters followed by various lower case letter subscripts which denote different grammatical aspects of the words that are connected through the linkages. For example, it uses ‘S’ link with various subscripts like ‘Ss’, ‘Sp’, ‘Sp*i’, ‘SX’, etc. to denote various features such as number, person, etc. of the verb and its subject [119]. The relations in Link parser are not directional. Hence information regarding the head and the dependent is also not available in LGP. Figure 4.6 will give you an idea about its dependencies:



```
                    +----Js---+
 +-Ss-+-Pg*b-+--MVp-+  +--Ds--+
 |    |      |      |  |      |
she is.v staying.v in the hostel.n
```

Figure 4.6: LGP linkages

Notice that the labels and the way they are approaching is very different and the dependency arcs are not directed. However, the decision of arcs is often similar to CLEAR.

We have briefly described only three parsers here, but our objective is to develop a dependency schema which can map any English parser.

4.3 Anusāraka Dependency Schema and the Motivation behind It

Before delving into our mapping scheme, it is pertinent to describe the grammatical framework that we have used as insights for developing these mappings.

The relations described by Pāṇini are syntactico-semantic in nature. These relations hold between nominal and verbal entities rather between their inflected forms. A nominal and verbal entity inflects to encode its relationship with other nominals or verbals in a sentence [116]. To proceed further, we have to answer the following question “How are the dependency relations encoded?”. In other words:

"What are the linguistic mechanisms that languages use to encode the grammatical relationships in a sentence?". Let us explore this through some examples.

(30) a. Mārjārāḥ mūṣakān mārayanti
cat.3.PL,NOM rat.PL,ACC kill.3,PL,PR
'Cats kill rats.'

b. Mūṣakān mārjārāḥ mārayanti

c. Mārjārāḥ mārayanti mūṣakān

d. Mārayanti mārjārāḥ mūṣakān

e. Mūṣakān mārayanti mārjārāḥ

f. Mārayanti mūṣakān mārjārāḥ

The words in Sanskrit example in 30a can be ordered in any possible combination as shown from 30b to 30f, still, they convey more or less the same meaning. But in English, changing the order of the subject and object as in "Rats kill cats" and "Cats kill rats" changes the meaning of the sentence all together.

What is it that gives Sanskrit such a greater degree of freedom of ordering the words which is not in English? The answer is the richer morphology of Sanskrit. All the words in example 30a have explicit morphemes to mark the desired information. Let us illustrate it in detail:

In example 30a, the word, *mūṣakān* explicitly presents itself as a theme/patient of the action through the case marker *-śas* which is realized as *-ān* in actual sentence.

The word, *mārayanti* is composed of the verbal stem *māraya* (to kill) and the verbal suffix *-anti*. The suffix *-anti* carries the information of third person, plural and active voice (*kartṛvācya*). It also carries the information that the doer (*kartā*) of this action will be denoted by a third person plural noun or pronoun.

The word *mārjārāḥ* is in nominative case. A nominative case is used when no explicit case has to be marked. As was just mentioned the inflected verb form *mārayanti* has already expressed that the 'doer' (*kartā*) of this action will be denoted by a third person plural noun or pronoun. So, its agreement with the verb *mārayanti* suggests that 'rat' (*mārjāra*) is the doer (*kartā*) of the action 'kill' (*mārayanti*).

Another mechanism of expressing relation information is through word order. In Sanskrit, compounds show this mechanism. For example, the compound *rājapuruṣaḥ* has a fixed order of the nouns *rājñaḥ* and *puruṣaḥ*. The relation between them is genetive, which in its paraphrase will occur as *rājñaḥ puruṣaḥ* or *puruṣaḥ rājñaḥ*. Once compounded, the genetive case marker is dropped and the order of the compound members gets fixed. The change in position of the participants also changes the meaning of the expressions. In Sanskrit, *rājapuruṣaḥ* means "a royal man" whereas *puruṣarājaḥ* means "a man who is also a king".

Based on the observations above, we can say that relational information in a sentence is encoded broadly by two means: (1.) by morphemes and (2.) by position or word order.

1. **By Morphemes:** The relation marking morphemes could broadly be classified into (a) nominal and verbal inflections (*sup* and *tiṅ vibhaktis*) and (b) derivational suffixes.

   (a) **Nominal and Verbal Inflections (*Sup* and *Tiṅ Vibhaktis*):** As we have already seen through the example 30, nominal inflections *sup vibhaktis* and verbal inflections *tiṅ vibhaktis* are the main relation marking elements in a sentence.

   (b) **By Derivative Suffixes:** The *kṛt* (primary affixes) and *taddhita* (secondary affixes) also code relational information. For example,

   (31) Rāmeṇa vedaḥ adhītaḥ
   Rama.INST.SG Veda.NOM,MASC.SG study.PT,3,MASC.SG
   'The Veda was studied by Rama.'

   In 31, the *kṛt* suffix *-ta*[1] suffixed to the verbal root 'to study' (*adhī*) denotes the theme/patient (*karma*) of the action. The relation between the theme/patient (*karma*) and action can be easily identified through their gender and number agreement. The *-ina*[2] suffix attached to the stem 'Rāma' denotes its doer (*kartā*) relationship with the action.

   The secondary derivative suffixes, the *taddhita* suffixes also come to denote some relationship of a participant with another. But one of the crucial differences between a secondary derivative and some of the primary derivatives is that they consume the other participant in relation to them. That is, between the two participants, only one remains. For instance, let us look at the example in 32:

   (32) Saindhavaḥ

   'Born in *sindhu* (ocean) or oceanic'

   In 32, the *taddhita* suffix *-a*[3] is attached to the nominal *sindhu* (ocean) to denote the *born in* relation. After the entire derivational process, the nominal *sindhu* (ocean) is realized as *saindhavaḥ* which means 'oceanic'.

   Though the *sup* and *tiṅ vibhaktis* and the *kṛt* and the *taddhita* suffixes mark relational information, the difference between these is that the *vibhaktis* come under inflectional morphology and the *kṛt* and the *taddhita* suffixes come under derivational morphology. *Vibhaktis* at one hand mark relations between two different words, the *kṛt* and the *taddhita* suffixes (except a few suffixes like *kta* and *ktavatu*) on the other hand mostly consume the other word and finally only one word is retained after the derivational process.

2. **By Position or Word Order:** As seen above, in Sanskrit compounds, relations between the compound members are marked through positions. Similarly, in positional languages like English,

[1] The primary suffix *-kta* is realized as *-ta* after deleting the marker *k*.

[2] The third case marker *-ṭā* is realized as *-ina*

[3] *The secondary derivative suffix -aṇ is realized as -a after deleting the marker ṇ.*

relations not only between/among the components of compounds but also between/among other participants in a sentence are also marked through positions.

We will deliberate on these two means in detail in section 4.7 and see how these can be applied to English for marking dependency relations.

The Anusāraka dependencies described in detail in Section 4.8, provide two types of dependency labels:

1. **Surface Oriented Dependencies:** These dependencies are based on syntactic information that language uses to encode the grammatical relationships between/among the words. For instance, in sentence 33 there would be a *kriyā-subject* relation between 'opened' and 'door'. The label *kriyā-subject* denotes that the word 'door' is the syntactic subject of the verb 'opened'. See labels in Figure 4.7.

   (33) The door opened.
   - daravājā khula.PT
   'daravājā khulā.'

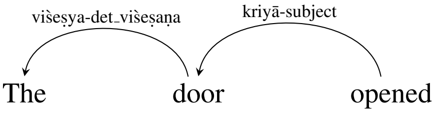


Figure 4.7: Anusāraka basic dependencies for 33

2. **Enhanced Dependencies:** The basis of these dependencies is mainly convenience. These dependency labels make the task of NLP like WSD etc. more convenient. For instance, syntactically the dependency relation between 'opened' and 'door' in sentence 33 and 34 is the same i.e. 'door' is the subject of the verb 'opened' in both the sentences. But the role of the 'door' is very different in these two sentences otherwise. This affects in disambiguation of the sense of the word 'opened' in these sentences. In 33, it means *khulanā* (to become open), whereas in 34, it means *kholanā* (to cause to open). A distinction between such syntactic structure makes the NLP tasks like WSD easy.

   (34) The door was opened.
   - daravājā.SG *AUX* khola de
   'daravājā khola diyā gayā thā.'

This study is motivated from arriving at a common representation for English to Indian languages, especially English to Hindi MT tool Anusāraka. Anusāraka is a full fledged translation system which uses dependency parse for analysis of the SL sentences. Various modules like Gender Number Person

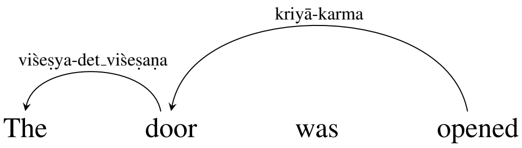


Figure 4.8: Anusāraka enhanced dependencies for 33

(GNP) agreement, Word Sense Disambiguation (WSD), reordering of the SL sentence according to TL sentence, etc. are some of the important tasks which require dependency parse labels as input facts in an MT system [89]. However, different parsers have different output schemes which differ in relation labels as well. Bringing the parsers' output into one uniform notation provides the system an ability to plug-in any of these parsers without modifying other subsequent translation modules. If we look at the Anusāraka architecture shown in Figure 4.9, we find that parsers come into picture at very early stage of the translation process. Mapping them into one uniform notation saves huge effort which is expensive as well as time consuming.

If we know the strengths and weaknesses of the parsers, we can select a parser that gives correct parse for the input sentence, so that the system produces best possible translation. Anusāraka's architecture allows plugging in various parsers for sentential analysis. Figure 4.10 shows Anusāraka dependency mapping module.

Thus, bringing parsers' output into a single uniform notation reduces the labor of re-writing the same logic for different parsers and it also results into generation of the relation labels that are more natural to the target languages.

4.4 Design Principles of the Proposed Schema

Our dependency labels are designed to provide a simple description of the grammatical relationships in a sentence that could be easily understood and effectively used by people who want to extract textual relations. The purpose is to build a system that helps Sanskrit scholars and common Indians to participate in the field of English to Indian language MT systems [22]. However, the dependency schema presented here is generic in nature and does have linguistic relevance for any other language as well.

The roots of dependency relation marking can be traced back to Pāṇini's kāraka theory [90], which provides many ingenious concepts for language analysis, which are 'universal' in nature. It gives a sound theory to identify the grammatical relations among the words in a sentence [15]. The importance of PG lies in the minute observations of Pāṇini regarding the information coding in a language [15]. In order to capture the way relational information is coded in English language at structural level, apart from the concepts like 'nominal inflections' (*sup*), 'verbal inflections' (*tiṅ*) and 'fully Inflected word form' (*pada*) described in Chapter 3, we will also look at the notion of *abhihita* (lit. expressed) from the Pāṇinian grammar.

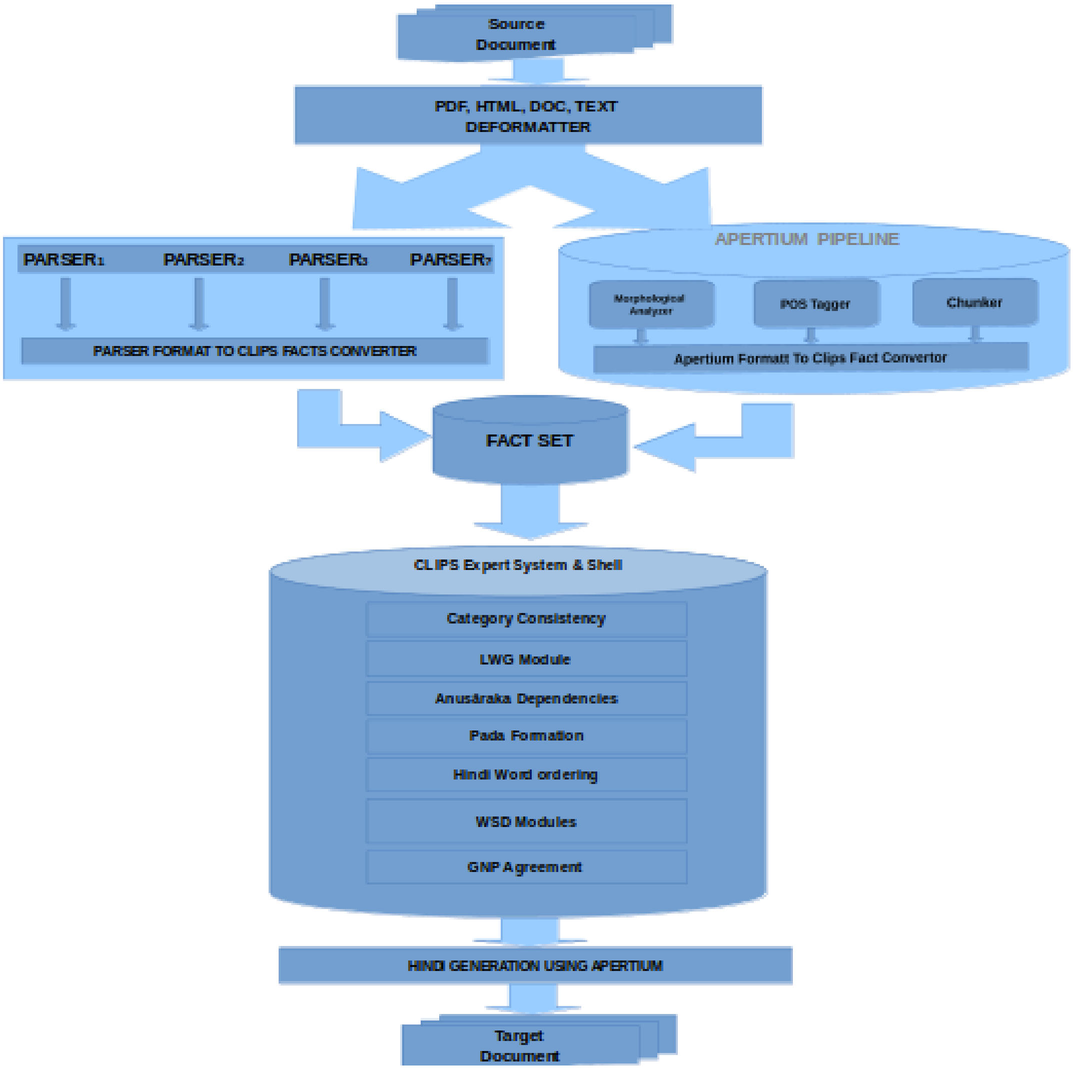


Figure 4.9: Anusāraka architecture

Pāṇinian grammar starts with semantics and ends with phonetic realization of the word, word group or the sentence [124]. It analyses a word as a combination of a root/stem (*prakṛti*) and an affix (*pratyaya*). It is both the root and the affix that jointly denote the meaning (*prakṙtipratyayau sahārtham brūtaḥ*) [109] (Lit. root and affix together convey the meaning). Pāṇini's grammar deals with morphology which is not separated but is interlinked with syntax and semantics [124]. This, in a way, helps capture "how languages encode information" and "how the information flows in language". For instance, English uses auxiliaries and prepositions as inflectional morphemes to express the grammatical relationships among the participants in a sentence. Most of the parsers connect them also with dependency labels. In our schema, we do not give any label to auxiliaries and prepositions rather we use them as features to encode the relations. We will explain it in detail in the forthcoming sections.

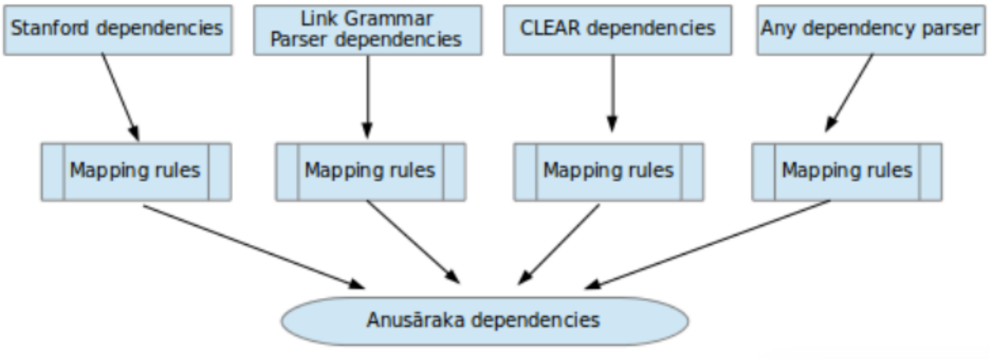


Figure 4.10: Anusāraka dependency mapping diagram

We believe in marking only that linguistic information which is more relevant from the point of view of information flow and information encoding, so that the information present in language string can be captured faithfully. We ignore marking the information that is less relevant from meaning point of view such as avoiding the information about singular and plural number of the participants as LGP does. Thus, the proposed schema is based on generalization of information flow in a language.

The design principles that we have followed throughout this schema can be summarized as:

- Distinguish relation holding elements from relation marking elements
- Try to give grammatical relations between/among content words only
- Use auxiliaries and prepositions as syntactic elements which encode the relational information
- Labels should be more informative and descriptive

4.5 Naming Convention of the Dependency Labels

Now, we will move to the naming convention that we have adopted for our dependency labels. The name ‘dependency’ is a suggestive term. Most of the current dependency schemes name the labels based on the dependent part. For example, take the label “nsubj” from Stanford dependencies. This label is given to the dependent part, the nominal subject of a relation that holds between a verb and its nominal subject.

In Pāṇini’s grammar, a sentence is treated as a series of modified-modifier relations (*viśeṣya-viśeṣaṇa-bhāva)*. In current dependency schemes, though the information regarding modified and modifier is available at the word level through the positions of the participants, it is not stated clearly in the labels itself. Most of the dependency schemes represent a label as a triplet of a relation between pairs of words. While doing this, they follow a convention that the first element is a dependency label, second element is the head and the third element is the dependent. So the head-dependent information of the participants is available in most of the dependency schemes, but it is not available in the relation labels itself i.e. by looking at the dependency label, one can not guess its head counterpart always.

Sometimes it is easy to infer the head information but it may not be clear always. For instance, if the labels are such as "nsubj" (nominal subject) or "dobj" (direct object), one can easily infer that the head of these relation labels would be a verbal element.

On the other hand, prepositional phrases are known to be ambiguous in their attachment in English and the label prep_in will not indicate its head explicitly. For example, take the sentence 35. One would find that the label "prep_in" occurs twice in this sentence, once between two nouns *boy* and *shirt* and secondly between a verb and a noun *sits* and *park*. The label "prep_in" does not evince anything about the heads of the labels. For translation purposes, capturing such differences is important. As the translations of both the prepositions are different as shown in 35.

(35) The boy in blue shirt sits in the park playing chess.
Ladakā **vālā** nīlī kamīja baiṭha.3.SG.PR **meṃ** udyāna khelatā huā chaisa
'Nīlī kamīja **vālā** ladakā chaisa khelatā huā udyāna **meṃ** baiṭhatā hai'

From the label "prep_in" one can not infer whether the head is a verb or a noun. For drawing such information while writing or extracting rules for language analysis, one needs extra help such as part of speech tags to make the information explicit. This makes the rules more cumbersome. Hence, we need a representation style which can state both parts of the relation labels clearly. Thus, our labels are couplets which contain the head and dependent information more explicitly.

In this schema, we have tried to make the dependency labels more explicit. A relation in this schema is represented by a triplet where the first part in every relation specifies the relation label that applies to its second and third parts. In this schema, every relation label has a hyphen (-) as a marker in it. Based on the hyphen, the relation name can be split into two sub-parts where the left sub-part always refers to the head and the right sub-part refers to the dependent. It follows the general view that the dependencies occur between two elements where one is the head and the other is a dependent. For instance, let us take the relation *(mother-daughter Marie Joliot*). In this relation, *mother-daughter* is the relation label and *Marie* and *Joliot* are its participants. In this label, the first sub-part *mother* corresponds to the first argument *Marie* and second sub-part *daughter* corresponds to the second argument *Joliot* as shown in Figure 4.11.

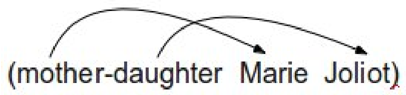


Figure 4.11: A diagram showing one-to-one mapping with the sub-parts of a relation name and its corresponding head and the dependent

Wherever possible, we have tried to name the labels in such a way that they reflect some syntactic or semantic phenomena in the sentence. Most of the labels in our schema are Sanskritized. Some of the labels are a combination of English and Sanskrit words like *kriyā-subject* where these labels project some syntactic or linguistic aspects of the language: hence, the labels become more informative. In cases where decision of any specific syntactic or semantic concept is not easy to identify, we stick to name the labels based on syntactic markers that encode the relations. For example, in sentence 36, the relation between the verb 'looks' and the noun 'mother' would be labeled based on the prepositional marker 'like' as *kriyā-**like**_sambandhī.* Similarly, in sentence 37, the relation between 'asked' and 'money' would be *kriyā-**for**_sambandhī.*

(36) She looks like her mother.

(37) She asked me for some money.

4.6 Mapping Dependency Labels

As mentioned above, our objective is to design a common representation of dependencies for various English parsers. Most of the dependency labels are mapped into Anusāraka schema just by changing the label names. For instance, Stanford parser and CLEAR parser give 'prt' relation between verbs and particles and LGP gives 'K' link. In our scheme, these labels are mapped into 'kriyā-upasarga' relation. In short, the mapping can be summarized in following four ways.

1. **One to One:** Labels like 'neg', 'poss', and 'tmod', etc. have one to one correspondence to the labels like *kriyā-kriyā_niṣedhaka, viṡeṣya-ṣaṣṭhī_viṡeṣaṇa* and *kriyā-kālavācī*, etc. For comparison, see table 4.1.

2. **One to NONE:** Some labels like 'aux' and 'possessive' have no mapping in our schema. Because, as mentioned before, 'aux' (which stands for auxiliaries) and 'possessive' (which stands for possessive marker), etc., are either function words or syntactic entities which are used as a device to express the semantic connection between content words.

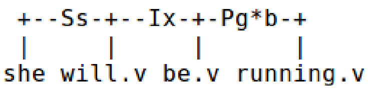
3. **One to Many:** Also one label is mapped into many because more refined information is required for translation. For example, most English parsers give equal treatment to all types of adverbial modifiers. In our schema, we give them four different labels based on the following four different classifications:

   - **Manner Adverbs:** These adverbs talk about how an action happens. For example, 'She ran quickly', in this sentence, 'quickly' is a manner adverb which says that the action of running happened at a high speed.
   - **Adverbs of Degree:** These adverbs show the degree or intensity of adjectives and other adverbs. For example, in this sentence 'Rama is a very intelligent boy', the adverb 'very'

shows the intensity of the adjective ‘intelligent’ and in ‘She runs very fast’, the adverb ‘fast’ tells about the manner of the action of running and the adverb ‘very’ tells about its intensity.

- **Adverbs of Time:** An adverb of time tells us ‘when’ something is done. For instance, ‘now’, ‘then’, ‘soon’ etc., tell us about the time of the action. For example, in the sentence ‘Do it here now’, the adverb ‘now’ describes the present moment of the action.
- **Adverbs of Space:** An adverb of space tells us about the locus of the action of something. For instance, ‘here’, ‘there’, ‘below’ etc. tell us about the locus of the action. For example, in the sentence ‘Do it here now’ the adverb ‘here’ describes the location of the action.

4. **Many to One:** Sometimes two or more labels in other parser sachems map into a single label in our schema. For example, see Figure 4.14 and Figure 4.16, repeated here as Figure 4.13 and Figure 4.12 for convenience. Figure 4.12 is a Link Grammar parser’s dependency parse tree. It shows three different labels among the words ‘She’, ‘will’, ‘be’ and ‘running’. Figure 4.13 is a parse produced using our schema. In this, the auxiliaries have been grouped with the verb, hence it shows only one label between the verb and its subject. Here the labels ‘Ss’, ‘Ix’ and ‘Pg*b’ are mapped into ‘kriyā-subject’.



```
+--Ss-+--Ix-+-Pg*b-+
|     |     |      |
she will.v be.v running.v
```

Figure 4.12: LGP dependencies

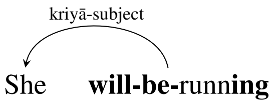


Figure 4.13: Anusāraka dependencies

Table 4.1 shows mapping of some major relation labels to our labels[4]. In the table, the symbol * after a label implies that zero or more subscripts get added to the label when it occurs in actual sentences. Labels followed by the † imply more than one labels in that particular scheme correspond to a single label in other schemes. Labels followed by the ‡ imply one label in other schemes corresponds to multiple labels in that particular scheme. Labels followed by the ± suggest that a chain of two or more labels corresponds to a single label in other schemes. The only hyphen (-) in a column means that the particular scheme does not have any corresponding label.

[4] For more details, the serious readers may look at the mapping programs that are distributed under GPL License with the Anusāraka English-Hindi MT system.

Stanford Dependencies	CLEAR Dependencies	LGP Links	Anusāraka Dependencies
advmod	advmod	EA, EC†	kriyā-kriyā-vis̀eṣaṇa†
agent	agent, pobj	MVp, J	kriyā-by-saṃbandhī
aux	aux	P*, PP	-
ccomp	ccomp	C*	kriyā-vākyakarma
cop	attr	O*t	subject-subject-samānādhikaraṇa‡
csubj	csubj	S*$^{\pm}$	kriyā-vākya-subject
dobj	dobj	O*	kriyā-object
iobj	iobj	Ox	kriyā-object-1
neg	neg	N	kriyā-kriyā-niṣedhaka
nsubj	nsubj	S*	kriyā-subject‡
nsubj	oprd	O*	object-object-samānādhikaraṇa
nsubjpass	nsubjpass	S*$^{\pm}$	kriyā-subject‡
pobj	pobj	J*	kriyā-preposition-saṃbandhī$^{\pm}$ †
possessive	possessive	YS, YP	-
poss	poss	D*	vis̀eṣya-ṣaṣṭhī-vis̀eṣaṇa
prep	prep	MV*	-
rcmod	rcmod	B*	vis̀eṣya-jo-samānādhikaraṇa
tmod	npadvmod	TT, Yt	kriyā-kālavācī
vmod	infmod	TO, I†	saṃjṅā-to-kṛdanta

Table 4.1: Showing a list of frequently occurring dependency labels with respect to Stanford dependencies, CLEAR dependencies, LGP links and Anusāraka dependencies.

4.7 Application of Pāṇinian Concepts to English

Sanskrit is a synthetic language where the root (*prakṛti*) and suffix (*pratyaya*) are combined to form one word (*pada*). English, on the other hand, is analytic in nature and uses word order (fixed positions) for encoding grammatical information. In this section, we discuss how the concepts of Sanskrit, that have been examined in Chapter 3, can be applied to English in order to give an account for its syntactic behavior.

4.7.1 *Pada* and Relation Encoding

In Sanskrit, *pratyaya*s include case morphemes and tense, aspect and modality (TAM) morphemes which are morphologically expressed on nouns and verbs respectively. On the other hand, English being an analytic language mostly uses free morphs for representing case and TAM information.

While Indo-European languages had inflections for eight cases in its nouns, English has lost them, except the genitive (possessive) marker "-'s" that still occurs as a bound morpheme. Following the Pāṇinian perspective, we treat 'preposition + noun' expressions (i.e. prepositional phrase) as *subanta padas* and 'verb base + TAM' constructions as *tiṅanta padas*. What it means is that a *pada* is treated as a single unit which depending on the situation, behaves as a head or as a dependent in a relationship. The auxiliaries and prepositions act as relation markers not as relation holders. Most parsers, however, treat auxiliaries and prepositions as independent tokens, hence, they assign dependency labels to them as well.

So, the concept of *pada* in PG, as described in Chapter 3, carries the information of the relations that different words in a sentence have with each other. In a *pada*, the inflectional suffixes, the *vibhaktis* express the relationships of the nominal and verbal entities with other *padas* (inflected words).

This section deals with the structural differences of the two languages and shows how conceptually these two languages are treated alike in Anusāraka MT system.

4.7.1.1 Treatment of Auxiliary Verbs

The auxiliaries (*tiṅ*) along with the main verb constitute a 'semantic unit' which is called a *tiṅanta pada*. The auxiliary verbs express Tense, Aspect and Modality (TAM). Since auxiliaries and verbs occur separately in English, we form a Local Word Group (LWG) [43, 13, 133] in order to represent *tiṅanta pada*. Since the relation between *tiṅ* and verbal root is not that of a modifier-modified, we propose that it is not necessary to mention the internal syntactic function (relations) between the verb and its auxiliaries [13]. Their function basically corelates with morphology. Therefore, the word group *'will be running'* in 38 is treated as a single unit, a *tiṅanta pada* in our schema. The *tiṅanta pada* then directly has a relation with the word 'she' as shown in Figure 4.14. Stanford and CLEAR dependencies and LGP, on the other hand, show three dependency relations among the words in 38 as shown in Figure 4.15 and 4.16 respectively.

(38) She will be running.

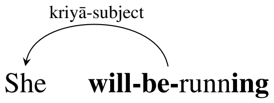


Figure 4.14: Anusāraka dependencies

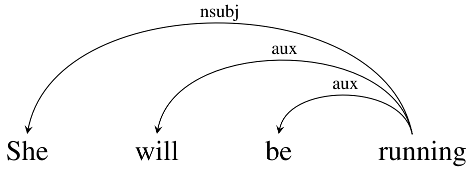
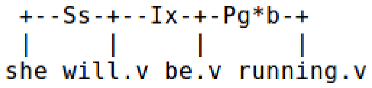


Figure 4.15: SD/CLEAR dependencies



```
 +--Ss-+--Ix-+-Pg*b-+
 |     |     |      |
she will.v be.v running.v
```

Figure 4.16: LGP dependencies

4.7.1.2 Treatment of Prepositions

Recall that prepositions are the equivalents of the *sup* suffixes which form *subanta pada* in Sanskrit. They denote grammatical relationships with other words in a sentence. Bharati et al. (1998) have pointed out that English codes the *kāraka* relations in positions as well as through prepositions.

Some parsers treat prepositions as function words while others as content words. For example, LGP treats prepositions as content words [17]. Our approach follows the principle of having direct relations with the content words. We treat the prepositions as relation marking morphemes. They correlate with morphology. They are not relation holders. Therefore, the prepositions become part of the relation name in our schema. In Stanford's basic typed dependencies, prepositions appear as dependents of the verbs and nouns, whereas in Collapsed and Collapsed CC-processed dependencies the prepositions are joined with the dependency label 'prep' [47]. For example, let us take the sentence 39 and see its dependencies generated by these parsers. in Figure 4.17 to Figure 4.21

(39) He runs in the garden.

LGP and CLEAR parsers treat prepositions as Stanford's basic dependencies (see Figure 4.17) except that LGP uses the labels 'Mvp' and 'Js' instead of the labels 'prep' and 'pobj' (see Figure 4.20). The

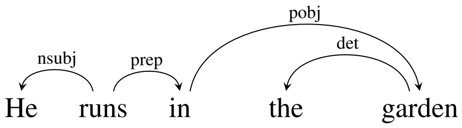


Figure 4.17: Stanford's basic dependencies

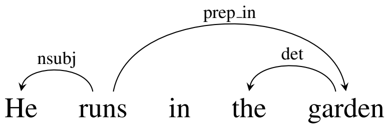


Figure 4.18: Stanford's Collapsed and Collapsed CC-processed dependencies

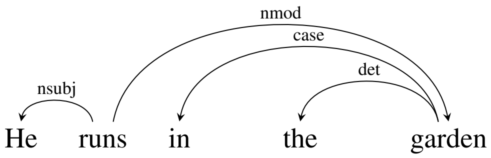
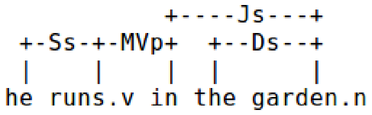


Figure 4.19: Universal Stanford dependencies



```
          +----Js---+
 +-Ss-+-MVp+  +--Ds--+
 |    |    |  |      |
he runs.v in the garden.n
```

Figure 4.20: LGP dependencies

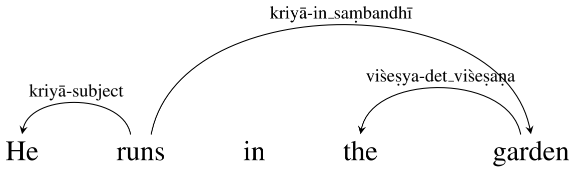


Figure 4.21: Anusāraka dependencies

taxonomy of grammatical relations proposed by Marneffe et al. (2014) treats prepositions as a dependent of the noun they are attached to (see Figure 4.19).

The complex prepositions such as 'along with', 'due to', 'out of', 'in front of', 'on account of' etc. function like an ordinary one-word preposition, hence we treat them also, similar to any other one-word preposition. Stanford treats some of the complex prepositions such as *because of, instead of, due to* as multi-word expressions (MWEs) [47], and some others similar to a single word prepositions. In our schema, all prepositions, single word or multi-word, are treated alike as shown in example sentence 40 and its dependency tree in Figure 4.22.

(40) The vast majority brought family along with them.

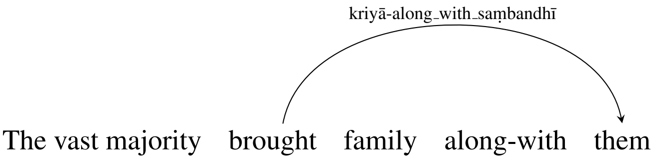


Figure 4.22: Anusāraka dependencies for complex prepositions

The treatment of prepositions as part of relation-labels brings the schema closer to *kāraka* theory.

4.7.1.3 Treatment of Genitive Morpheme *"'s"*

Genitive case in English is realized in two ways: (i) by a post-nominal *of*-phrase, and (ii) by affixation of the genitive morpheme *'s* to a pre-nominal NP (noun phrase) [65]. The genitive morpheme *'s* is also referred to as Saxon genitive [65].

Most of the present dependency parsers mark the edge between the genitive morpheme *'s* and the noun it attaches to with *possessive* or similar relation, and assign the relation *poss* to the edge between the head of an NP and its possessive determiner, or a genitive *'s* complement [47].

In case of pronouns, the genitive case is indicated by possessive pronouns like *my, your, his, her, its, our* and *their*. No parser splits the possessive pronouns into two parts as they do with the Saxon genitive NPs such as *John's*.

In our schema, the genitive morpheme *'s* is not separated from the nominal it case marks because the whole word form is considered as a *pada*. In our schema, the dependency relation holds between the head of an NP and its possessive determiner or genitive modifier. See Figure 4.23 to 4.25. Haegeman and Guéron (1999) also suggest that grouping possessive pronouns such as *his* and Saxon genitive NPs such as *John's* is a natural step: both are essentially genitive forms of a related NP (*he* and *John* respectively).

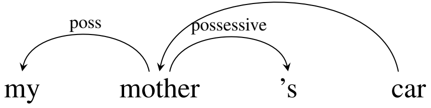


Figure 4.23: Stanford dependencies for Saxon genitive singular

An *of*-phrase is treated similar to any other prepositional phrases.

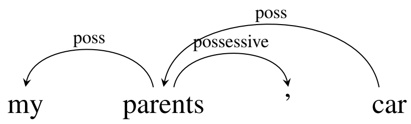


Figure 4.24: Stanford dependencies for Saxon genitive plural

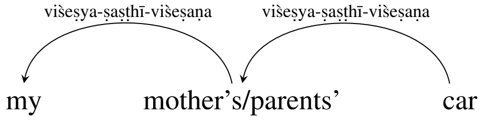


Figure 4.25: Anusāraka dependencies for singular and plural Saxon genitive

Thus, auxiliary verbs, prepositions, infinitive marker ‘to’, and Saxon genitive *’s* are collapsed out of our representation schema making the schema more useful for relation extraction and shallow language understanding tasks.

4.7.2 The Concept of *abhihita*: ‘Subject’ and ‘Object’ vs *kartā* and *karma*

As mentioned above, the relations are also marked through the positions of the words. The position could be either the position of the components in compounds or any other position such as the subject and object positions. In this section, we discuss the subject and object positions from Pāṇinian view point.

The subject and object relations as attested by other dependency parsers come closer to the Pāṇinian concept of *abhihita* ‘Expressed’. The concept is more syntactic in nature unlike the notion of *kartā* and *karma* which are semantic in nature.

The notion of ‘subject’ is often mistaken as *kartā* (doer). Even though sometimes it does correspond to *kartā*, factually the subject in English is not similar to *kartā* in Sanskrit. Let us examine it through sentence 41:

(41) Saḥ tān hanti
He them kill.PR,3,SG
‘He kills them’

The subject ‘he’ in English translation in 41 corresponds to *kartā*, which ends in *prathamā vibhakti* (nominative case). The original *vibhakti* for *kartā* as defined by Pāṇini is *tṛtīyā* (third case) (A. 2.3.18). According to the notion of *abhihita*, the *kartā*, ‘he’ in 41 is not assigned *tṛtīyā vibhakti* because the verb which is in *kartṛvāchya* (active voice) has already expressed the *kartā* of the action *kill*. Therefore *kartā* remains in *prathamā vibhakti*. In Sanskrit, as has already been said, *prathamā vibhakti* is used to express only *prātipadikārtha* (nominal stem meaning), *liṅga* (gender), *parimāṇa* (measure) or *vacana* (number). According to (A. 2.3.18) the *kartā* has to remain un-expressed in order to be assigned *tṛtīyā* vibhakti, .

Let us transform the sentence in 41 into passive voice as shown in 42 and see whether the subject and the object of these examples correspond to the *kartā* and *karma* respectively.

(42) They were killed by him
They.3,PL,NOM kill.3,PL,PT he.INST,SG
'Te tena ahanyanta'

In passive construction also, 'he' (him) is *kartā* (doer) of the action *kill* as it was in active voice in 41, although it does not occupy subject position in 42. Figure 4.26 and 4.27 show surface level dependencies for sentence 41 and 42 respectively and Figure 4.28 and 4.29 show surface level dependencies for their respective Sanskrit translations.

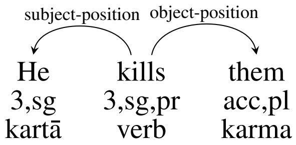


Figure 4.26: Surface level dependency information for English sentence in 41

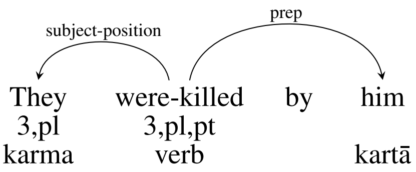


Figure 4.27: Surface level dependency information for English sentence in 42

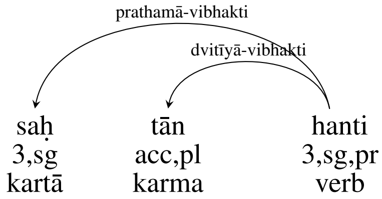


Figure 4.28: Surface level dependency information for Sanskrit sentence in 41 from Pāṇinian perspective

The subject 'they' in English sentence in 42 is translated as *te* in Sanskrit which is *karma,* ending in *prathamā vibhakti*. The original *vibhakti* for *karma* as defined by Pāṇini is *dvitīyā*. In these examples, the information regarding *karma* is expressed by the post-verbal affixes and that is the subject. The agreement between the verb and the *karma* is its surface realization. Thus, the subject and the object in English are based on position and can not be mapped into *kartā* and *karma*. But subject can be mapped to

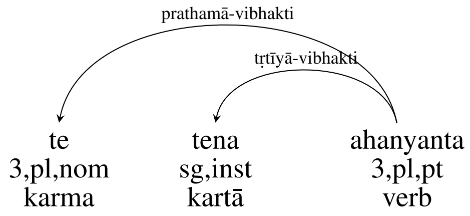


Figure 4.29: Surface level dependency information for Sanskrit sentence in 42 from Pāṇinian perspective

abhihita. Whether the subject corresponds to *kartā* or *karma* in a given sentence can be inferred through the verbal suffixes, the active or passive *tiṅ* suffixes. In active voice, the subject generally corresponds to *kartā*, and the object corresponds to *karma*. Thus, the subject and object are the syntactic relations, whereas *kartā* and *karma* are the syntactico-semantic relations [11].

Another interesting fact to be noticed is that in Sanskrit, if the *kartā* or *karma* is expressed by *tiṅ* suffixes, then it takes *prathamā vibhakti*. Similarly, the element that has been expressed by English verbal suffixes, comes in subject position.

Anantpur (2009) argues that English lacks an explicit morphological formative for accusative marking. Rather it codes the information to indicate grammatical relations of subject and object by their positions. For such reasons, we do not map 'subject' and 'object' into *kartā* and *karma*. Also, English is a positional language, i.e. English has relatively fixed word order with word positions containing some grammatical information. Since the notion of 'subject' also contains 'positional' information, it is useful to retain this notion as such. Therefore, in our schema we have decided to keep 'subject' and 'object' also as relation labels [22].

Also, at present, we are better able to extract the surface level information. At second level, we might do deeper analysis to extract the kāraka level semantics.

As a conclusion, we can say that when the information is marked through *tiṅ* (finite verbal) suffixes then Sanskrit uses *prathamā vibhakti* to mark the *pada* whereas English uses subject position as a *vibhakti*.

4.7.3 Treatment of Copular Verbs

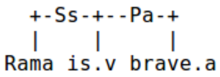
A copula verb is a non-action verb. Traditional grammarians use the term *copula* to refer to the verbs with little semantic content. *Be (is, are, was, were), look, seem, appear, feel, smell, sound* etc. are some of the examples of copula verbs. A copula verb serves to link a subject with a predicate [110], usually a subject complement that refers to the subject. For instance, the subject in sentences in 43 is linked with the predicates using the copula verbs 'is' and 'looks'.

(43) a. Rama is brave.

b. Rama is a king.

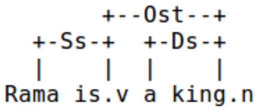
c. She looks beautiful.

Many theorists consider that copular verbs do not describe actions being performed, that is why they treat them as auxiliary verbs, even when the complements of these verbs are adjectives or predicative nouns. Therefore the head of such sentences is no longer a verb. For example, Stanford parser considers a copula as a dependent of its complement [47]. LGP and CLEAR parser view them as the head of the sentence. LGP treats sentences 43a and 43b differently as shown in Figure 4.30 and 4.31.



```
  +-Ss-+--Pa-+
  |    |     |
Rama is.v brave.a
```

Figure 4.30: LGP dependencies for sentence 43a



```
        +--Ost--+
  +-Ss-+  +-Ds-+
  |    |  |    |
Rama is.v a king.n
```

Figure 4.31: LGP dependencies for sentence 43b

Stanford parser though gives same treatment to both these sentences but the labels and their heads are different with respect to LGP and CLEAR parser because it does not treat copula verb as head of the sentence. See Figure 4.32.

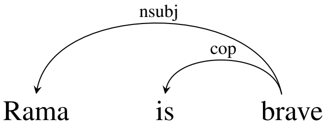


Figure 4.32: Stanford dependencies for the sentence 43a

CLEAR parser though makes copular verb head of the sentence but it treats 43a and 43b differently. See Figure 4.33 and 4.34.

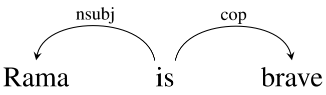


Figure 4.33: CLEAR dependencies for the sentence 43a

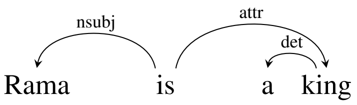


Figure 4.34: CLEAR dependencies for the sentence 43b

Traditional Indian Grammarians' Views Regarding Copula Verbs

Indian grammatical tradition treats copular verbs exactly in the same manner as other verbs like *pac* (cook), *kṛ* (do) etc. as what comes into being and what is brought into being [97]. Following are key points based on which copular verbs are treated similar to the other verbs.

1. Yask counts the verb *as* (be) among the six modes[5] of *bhāva*, namely *kriyā* (action) [71, 135].

2. "When a certain thing is maintaining itself by itself, it is referred to by the finite verb form *asti* (be)" [97]. In 43, the activity of self-maintenance is assumed by the agents.

3. Finite verb forms convey action, and at the same time, they show time divisions. Similarly, by the use of the copula verbs such as 'is', 'are', 'was', 'were' etc. also three time divisions are manifested [97]. Therefore, we consider copular verbs as any other verbs and make them heads of the sentences.

4. Patanjali states the property of an action to exclude other actions[6]. In the sentences in 43, the copula verbs rule out the possibility of other verbs. Therefore, they should be treated as other verbs.

Based on the above said reasons, we have treated the copula verbs as any other verbs. Figure 4.35 shows Anusāraka dependencies for sentence 43a.

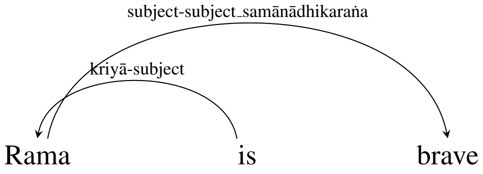


Figure 4.35: Anusāraka dependencies for the sentence 43a

[5] Ṣaḍbhāvavikārā bhavantīti Vārṣyāyaṇiḥ. (1) *Jāyate* 'genesis', 'is born', 'comes into being', (2) *asti* 'existence', 'is', (3) *vipariṇamate* 'alteration', 'changing', 'undergoes modification', (4)*vardhate* 'grows' 'increases', (5) *apakṣīyate* 'decay', 'diminishing', and (6)*vinaśyati* 'perishes', 'destruction' [135].

[6] *kriyāyāḥ kriyā nirvartikā bhavati*

4.7.4 Treatment of Relative Pronouns

A relative pronoun introduces relative clause. It has the same referent in the main clause of a sentence that the relative clause modifies. Identity of a relative pronoun depends on its referent.

Generally a verb is considered to be the head of a clause because it is the verb that connects the clause with other words in the sentence. But in case of the relative clause it is the relative pronoun that connects the clause with its referent. The relative pronoun takes the gender and number information from its referent. It agrees with its antecedent in gender, number and person. The main verb of the relative clause does not agree with relative pronoun or the antecedent. For example, in 44, the relative pronoun 'whose' is singular because it has agreement with a singular referent 'the child'. The verb 'are' in the relative clause is plural because it agrees with its plural subject *parents*.

(44) Fortunate is the child whose parents are learned.

Hence instead of the verb of the relative clause it is the relative pronoun that has direct relation with its referent in our scheme.

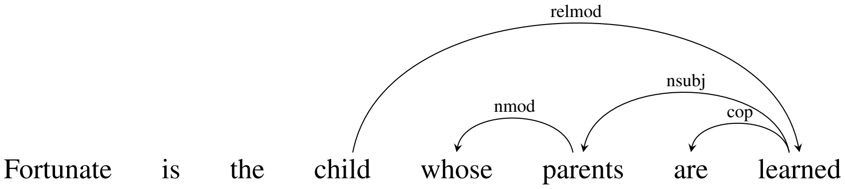


Figure 4.36: Stanford dependencies for the sentence 44

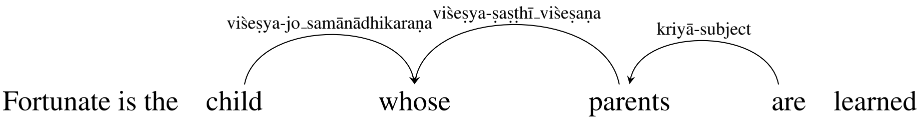


Figure 4.37: Anusāraka dependency parse trees for the sentence 44

It differs from other *samānādhikaraṇas* in terms of *vibhakties*. In other *samānādhikaraṇas* both the *samānādhikaraṇas* and its modified have same *vibhakti*. But the relative pronouns can have different *vibhakties*.

4.7.5 Treatment of Causative Constructions

Different languages mark information differently. Within a same language also, words may have dual status [65]. For example,

(45) a. I **have** to cook dinner.

b. I do not **have** to cook dinner.

In 45, the boldfaced verbs are modal auxiliary verbs.

(46) a. I **had** my daughter cook dinner.

b. Did you **have** my daughter cook dinner?

In 46, the boldfaced verbs are called causative verbs. The subject of these verbs causes an event to take place [65]. The event here is: *'cause to cook'*. The behavior of *make, cause* and *get* in 47 - 50 is similar to the causative verb *have* in 46.

(47) a. I laugh.

b. She makes me laugh.

(48) a. I wrote an essay.

b. The teacher made me write an essay.

(49) a. The girl fed the child some fruits.

b. She made the girl feed the child some fruits.

(50) a. I bought a computer.

b. My children got me to buy a computer.

Notice that the causative derivation increases the valency of the verb. It adds one more argument to the argument structure of the verb. For example, the intransitive verb 'laugh' in 47a has one argument but in 47b it has two arguments; in 48a, the verb 'wrote' has two arguments; in 48b it has three; in 49a, the verb 'fed' has three arguments; in 49a, the verb 'fed' has three arguments; in 49b, it has four arguments.

While English uses the verb *make, have, get* or *cause* to express the causativization, Hindi uses *-ā* and *-vā* suffixes to the root to represent direct and indirect causation respectively [107]. For causatives, dependency parsers first make causal verbs: *make, have, get* or *cause* as the structural head of the dependency parse, but in Hindi these verbs are replaced with causative suffixes *ā* or *vā* which are then affixed to the non-causative or second causal verbs for causativization. The TAM or the *tiṅ*, which in English is attached to the first causal verbs, is transferred to the second causal verbs in Hindi. The subject of the first causative functions as a mediating agent in the corresponding second causative [72]. We name it *prayojya_kartā* [108]. It is also important to notice that the head verb of the complement clause becomes the head of the entire causative sentence in Hindi.

Since the verbs *cause, make, have* and *get* express causativization, we give the relation *kriyā-preraka_kriyā* (verb-causative-marker) between the causative and the non-causative verb. This relation label triggers the causativization of the main verb in Hindi where it is a morphological process. See the differences in dependency trees in Figure 4.38 and Figure 4.39.

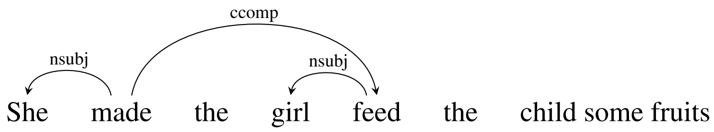


Figure 4.38: Partial Stanford dependency tree for 49b

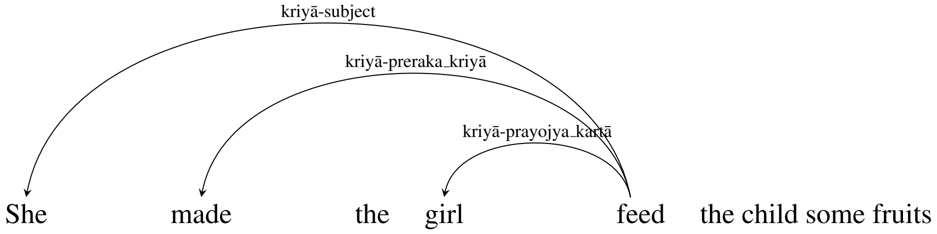


Figure 4.39: Partial Anusāraka dependency tree for 49b

4.7.6 Treatment of Coordinating Conjunctions 'AND' and 'OR'

Coordinating conjunctions conjoin two or more constituents of the same syntactic importance. Several approaches are used for coordination representation in dependency parse. The SDs make the leftmost conjunct the head of all other conjuncts and conjunctions. CLEAR parser makes the leftmost conjunct the head of the dependency and the conjunction is connected with immediate left conjunct. LGP connects each conjunct with the head and leaves the conjunction unlabeled. The basic Anusāraka dependencies represent relations with the conjunction, whereas in the enhanced Anusāraka dependencies, the relation label would be attached to all the conjuncts.

These constructions consisting of coordinating conjunctions correspond to the *dvandva* compound where two or more words connected with each other in the sense of *ca* (and) form a compound. We call it *quasi-dvandva* because in a *dvandva* compound the meaning of the word *ca* remains but not the word *ca* itself. For example, the *dvandva* of *Rāmaḥ ca Kṛṣṇaḥ ca* is *Rāmakṛṣṇau* (Lit. Rama-Krishnas) where both the *ca*s are dropped in the compound form. But in our examples, except the last *ca*, all other *ca*s are dropped.

(51) a. John, Mary and Sam laughed.
b. John, Mary or Sam laughed.

For such reasons, we name it 'quasi-dvandva'. We also call such constructions 'extended dvandva' or 'generalized dvandva' because we have extended it to the sense of *'vā'* (or) as well, whereas the original dvandva is prescribed only in the sense of 'and' (*ca*[7]).

[7] cārthe dvandvaḥ (A. 2.2.29) [129]

While marking the dependency labels, our approach makes the conjunctions head because it captures the meaning of the compound. All the components of it are available through the conjunctions.

For theoretical concerns we make the conjunction head or dependent of the relation and all the conjuncts get indirectly connected with the head or the dependent. But for practical concerns such as WSD, we give direct relation with each conjunct.

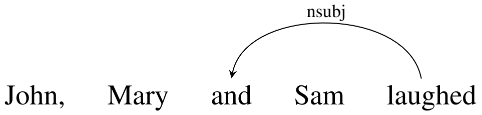


Figure 4.40: Anusāraka basic dependencies for 51a

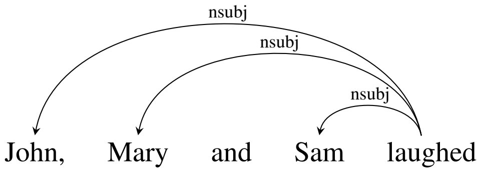


Figure 4.41: Anusāraka enhanced dependencies for 51a

Languages also have constructions where coordinating conjunction occures after each conjunct except the last one as shown in 52. In such verbose constructions, the last conjunction is made the head.

(52) a. John and Mary and Sam laughed.

b. John or Mary or Sam laughed.

4.8 Description of the Relation Labels

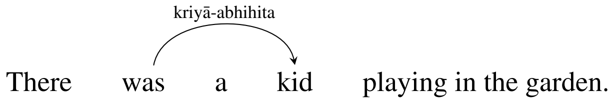
Our dependency labels can be classified into two classes: (i) intra-pada relations and (ii) inter-pada relations.

4.8.1 *Inter-pada* Relations

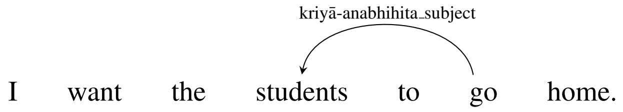
In this section, we will describe the inter-pada relation labels. These relation labels hold across the *subanta* and *tiṅanta padas*.

1. **kriyā-abhihita:**
The ‘subject’ in English is a pre-verbal position which is expressed through the subject verb agreement. Unlike any other subject, ‘there’ is the grammatical subject in expletive sentences. It can be seen by the fact that it inverts with the finite auxiliary: ‘Was there a kid playing in the

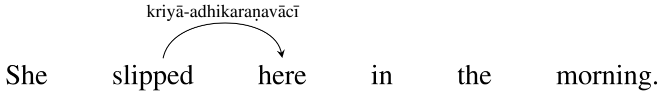
garden?'. In the example, 'there' functions as a place-holder for the logical subject, 'a kid', which occupies the complement position, but still the verb expresses its agreement with it. Therefore, instead of calling it a subject, we call it *abhihita* (expressed).

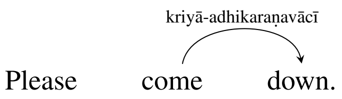


2. **kriyā-anabhihita_subject:**
   As mentioned earlier, in our approach subject is a pre-verbal position. The 'subject' in English corresponds to the concept of *abhihita* in Sanskrit. In other words, the subject is expressed by the verbal suffixes. That is why there has to be subject verb agreement for the sentence to be grammatical. In the examples such as the one given here, the subject 'students' is not being expressed by the verbal suffix in 'go', hence we call it 'anabhihita_subject' (unexpressed subject) of the verb 'go'.

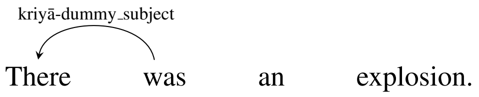


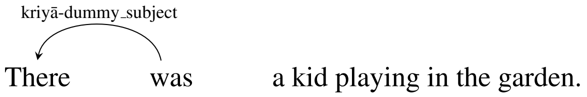
3. **kriyā-adhikaraṇavācī:**
   This relation label is used to show the relationship of the verbs with spatial modifiers like 'here', 'there', 'down' etc.



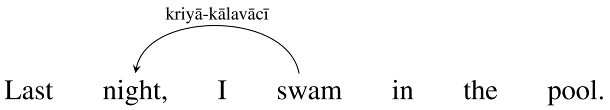


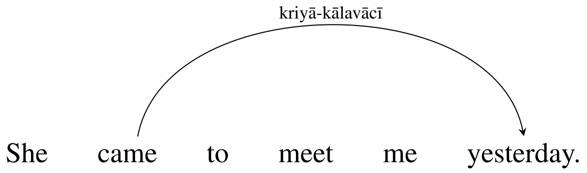
4. **kriyā-dummy_subject:**
   In our scheme, for something to be the subject of a verb, it has to be expressed (*abhihita*) by the verb. In English, it is seen through agreement. In a sentence where subject is not expressed by the verb but still it comes at the subject in the sentence, we call it 'dummy_subject'.



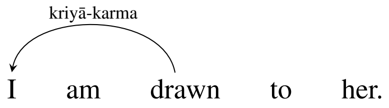


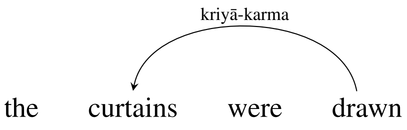
5. **kriyā-kālavācī:**
   This relation connects verbs with temporal expressions which do not have any overt preposition but they contain the meaning of the prepositions like ‘in’, ‘on’ etc.



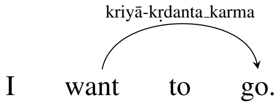


6. **kriyā-karma:**
   In enhanced dependencies, the label *kriyā-karma* connects a passive verb with its subject. In passive voice, the syntactic subject of the verb semantically stands for the theme/patient (*karma*) of the verb. This label helps in word translation disambiguation. At the basic dependency level, we also give the label ‘kriyā-subject’ to such constructions.



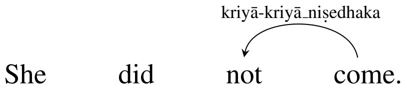


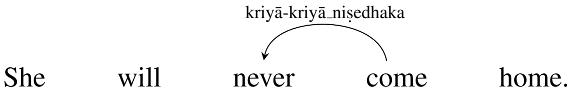
7. **kriyā-kṛdanta_karma:**
   A *kṛdanta* is a non-finite verb form. This relation label connects a verb with a *kṛdanta* that acts as a theme/patient/object of the verb.



8. **kriyā-kriyā_niṣedhaka:**

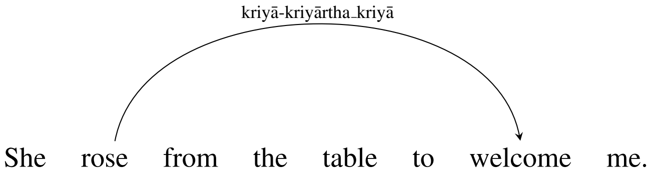
   This label is used to show the relationship of the verbs with negation modifiers.





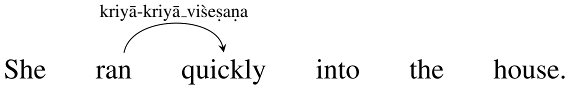
9. **kriyā-kriyārtha_kriyā:**

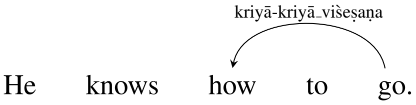
   The term *kriyārtha_kriyā* refers to the verb marked with infinitival 'to' when it denotes purpose. This label connects a verb with an infinitival verb when the 'to verb' means "in order to".



10. **kriyā-kriyā_viśeṣaṇa:**

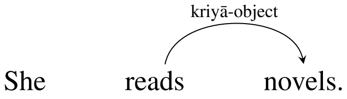
    This label connects verbs with the adverbs of manner. Manner adverbs tell us how something is done or how something happens.





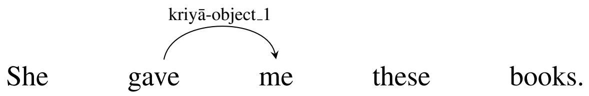
11. **kriyā-object:**

    This label connects a transitive verb with its accusative object.

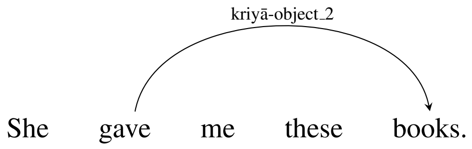


12. **kriyā-object_1:**

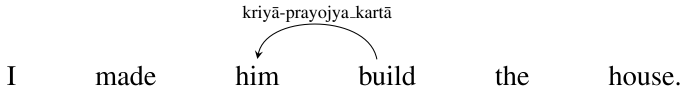
    This label relates a ditransitive verb with its indirect object. Since the indirect object occupies the first object position, it is named as 'object_1'. It is the dative object which corresponds to *sampradāna kāraka* in Sanskrit.

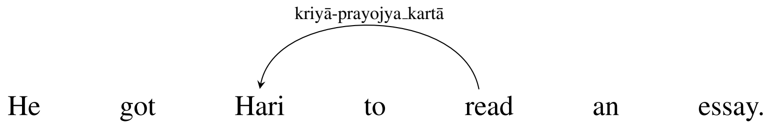


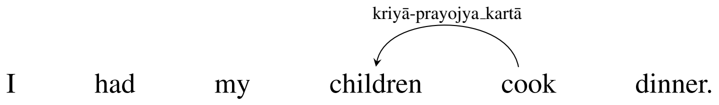
13. **kriyā-object_2:**
When the direct object of a ditransitive verb occupies second object position it is labeled as object_2. This relation label connects a ditransitive verb with direct object.



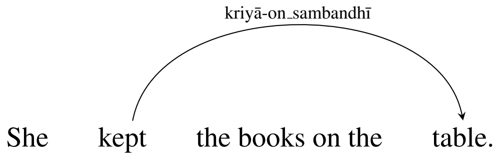
14. **kriyā-prayojya_kartā:**
In a causative construction, the causee (the first object of a verb) is called *prayojya_kartā*. The label *kriyā-prayojya_kartā* connects the second causal verb with the *prayojya_kartā*.

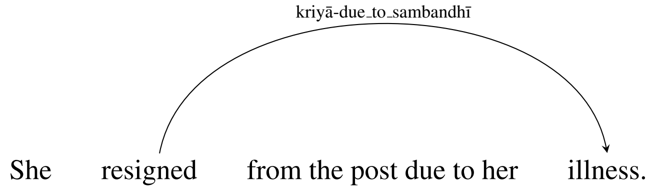




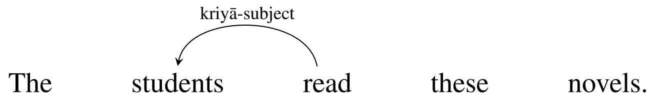


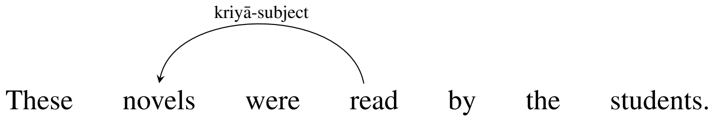
15. **kriyā-preposition_sambandhī:**
This label connects a verb with the head of its prepositional participant. In actual label in a sentence, the word ‘preposition’ is replaced with the prepositions ‘in’, ‘on’, ‘from’, ‘about’, ‘due to’, etc. that connect the verb with the prepositional objects. The prepositions do not have any label directly. For instance,



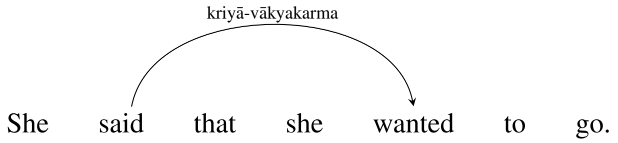


16. **kriyā-subject:**
    This label connects the verbs with their syntactic ‘subject’.

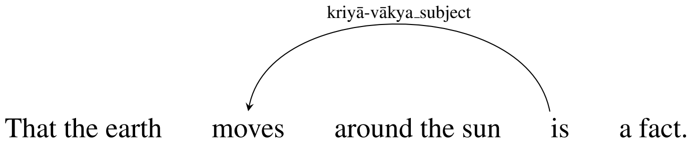


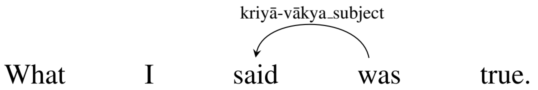


17. **kriyā-vākyakarma:**
    An argument clause is termed *vākyakarma* if it functions as an object of a predicate. Thus, the label *kriyā-vākyakarma* connects a verb with the head verb of its objectival argument clause.



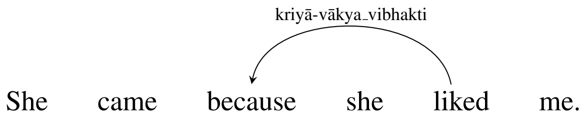
18. **kriyā-vākya_subject:**
    A clause is named as ‘vākya_subject’ when it stands as a syntactic subject to a verb. The relation ‘kriyā-vākya_subject’ holds between a verb and its clausal subject. Since verbs are taken to be the central point of a clause, the dependent of this relation is the verb of the subject clause.





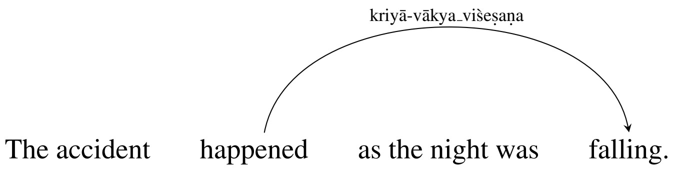
19. **kriyā-vākya_vibhakti:**

    A *vākya_vibhakti* is a subordinating conjunction (such as although, because, while, etc.) that introduces a clause. Similar to a preposition, a subordinating conjunction connects a clause to a verb. This label connects a verb in a clause with a subordinating conjunction which has introduced the clause.



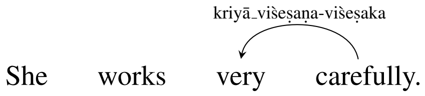
20. **kriyā-vākya_vis̀eṣaṇa:**

    The main verb of a clause is marked as a *vākya_vis̀eṣaṇa* if the clause acts as a modifier of a verb. This label connects a verb with the modifier clause.



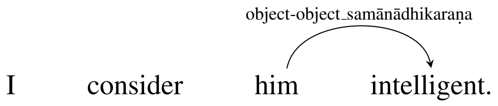
21. **kriyā_vis̀eṣaṇa-vis̀eṣaka:**

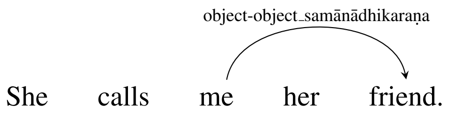
    This label is used to connect manner adverbs with the intensifiers that modify them.

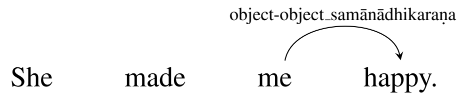


22. **object-object_samānādhikaraṇa:**

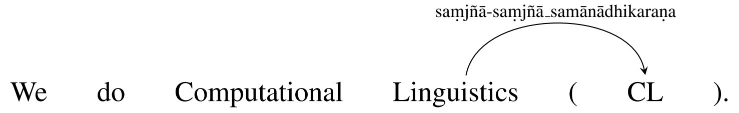
    Samānādhikaraṇa means ‘having the same locus’. When two or more entities refer to the same object, they stand in *sāmānādhikaraṇya*, ‘co-reference’ relationship. This label connects direct object of a verb with the *object_samānādhikaraṇa* ‘object complement’ because they have the same locus as object. An *object_samānādhikaraṇa* either renames the direct object or states what it has become.



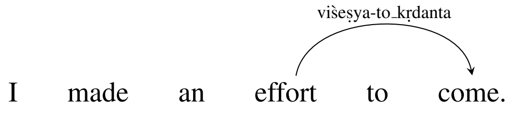




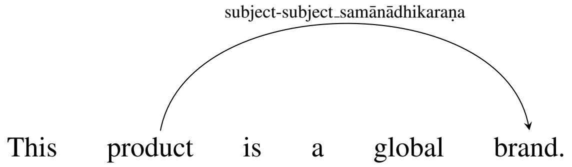
23. **saṃjñā-saṃjñā_samānādhikaraṇa:**
Saṃjñā means noun. *Samānādhikaraṇa* means ‘having the same locus’ or ‘co-reference’. So, *saṃjñā_samānādhikaraṇa* is a noun that has the same locus with another noun i.e. when two or more entities refer to a same object, they stand in *sāmānādhikaraṇya*, ‘co-reference’ relation. This relation label connects a noun with its appositive modifier.

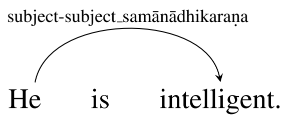


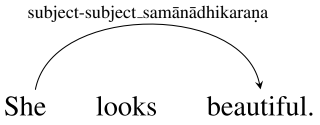
24. **viŝeṣya-to_kṛdanta:**
This label connects nominals with their infinitival modifiers, the ‘to + verbal base’ constructions. A *kṛdanta* is a non-finite verb form which is restricted only to an infinitival verb by the infinitival marker ‘to’.



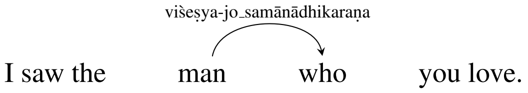
25. **subject-subject_samānādhikaraṇa:**
Samānādhikaraṇa means ‘having the same locus’. As we know, when two or more entities refer to a same object, they stand in *sāmānādhikaraṇya*, ‘co-reference’. This label connects the syntactic subject of a verb with the *subject_samānādhikaraṇa* ‘subject complement’. Thus, a *subject_samānādhikaraṇa* can be a noun, pronoun or adjective that follows a linking verb. The subject complement either renames the subject or defines it in some way.

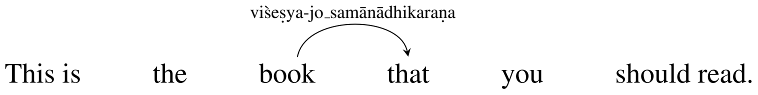


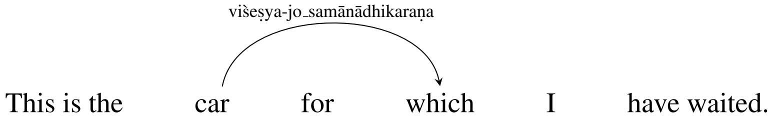




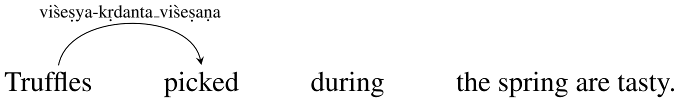
26. **vis̀eṣya-jo_samānādhikaraṇa:**
Samānādhikaraṇa means ‘having the same locus’. As said before, when two or more entities refer to a same object, they stand in *sāmānādhikaraṇya*, ‘co-reference’ relation. This label connects nouns with their relative pronouns.

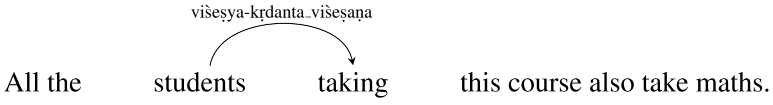




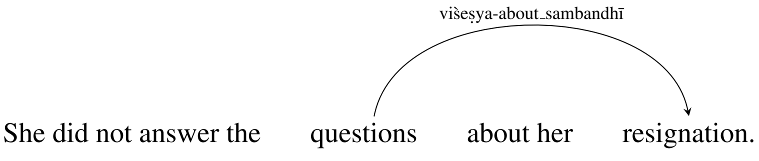


27. **vis̀eṣya-kṛdanta_vis̀eṣaṇa:**
A *kṛdanta* is a non-finite verb. This relation occurs when a non-verbal element is modified by a *kṛdanta*.

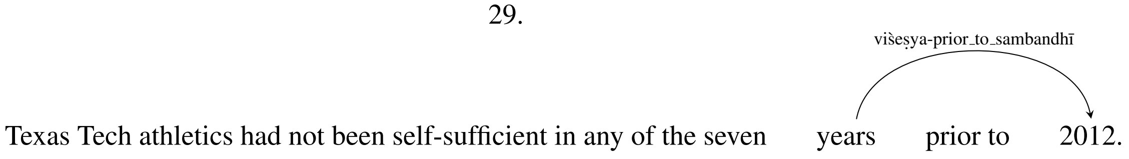




28. **vis̀eṣya-preposition_sambandhī:**
This label connects a nominal with the head of its prepositional phrase modifier. In actual labels, in a sentence the word ‘preposition’ is replaced with the *sup vibhaktis* ‘in’, ‘on’, ‘from’, ‘about’, etc. that connects a nominal with another nominal. For instance,

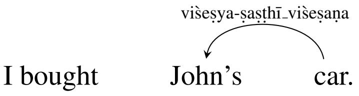


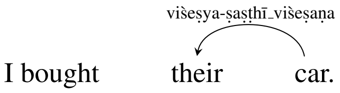
29.


viŝeṣya-prior_to_sambandhī
Texas Tech athletics had not been self-sufficient in any of the seven years prior to 2012.

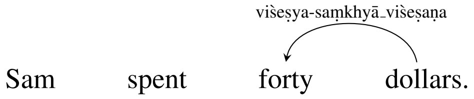
(borrowed from [44])

30. **viŝeṣya-ṣaṣṭhī_viŝeṣaṇa:**
This label connects possessive modifiers with their head nouns.

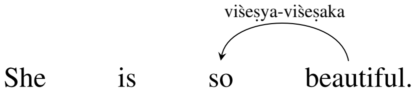




31. **viŝeṣya-saṃkhyā_viŝeṣaṇa:**
A *saṃkhyā_viŝeṣaṇa* is a numeric modifier that gives quantity of a noun. So, this label connects a noun with its numeric modifier.



32. **viŝeṣya-viŝeṣaka:**
This label connects adjectives with intensifiers.



4.8.2 *Intra-pada* Relations

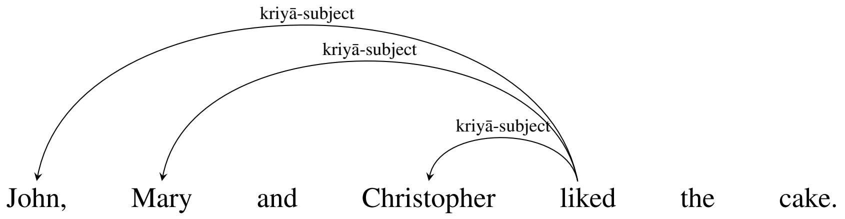
In this section, we will give a description of the relation labels that hold within the members of a *pada/samasta-pada*.

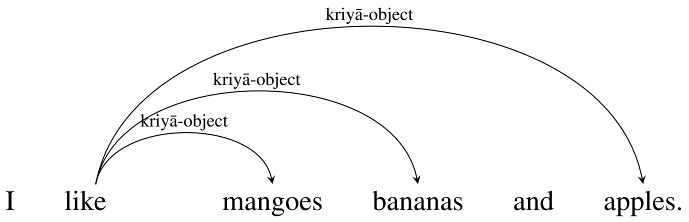
1. **conjunction-components:**
This label connects coordinating conjunction with the conjuncts in coordination. The head of this relation is a coordinating conjunction and the modifiers are the conjuncts connected by the coordinating conjunction. This is an enhanced level dependency. When the coordinating conjunction gets a label like 'kriyā-subject', 'kriyā-object' etc., this label lists out the conjuncts connected with the coordinating conjunction.

John, Mary and Sam ate cake. (conjunction-components and John Mary Sam)

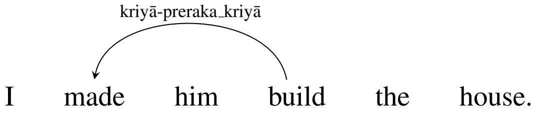
She met John, Mary and Sam. (conjunction-components and John Mary Sam)

In basic dependencies, each conjunct gets a label according to its grammatical relationship in the sentence.

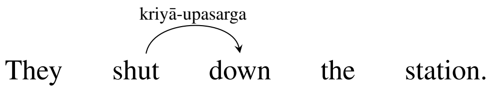


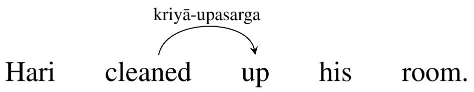


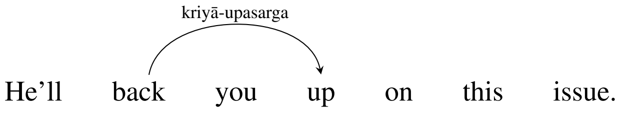
2. **kriyā-preraka_kriyā:**
   In English, causative verbs (make, have, get etc.,) loose their own meaning and add an extra meaning "to convince to do something" or "to trick someone into doing something" to other verbs; but, in many languages including Hindi and other Indian languages, causativization is marked through suffixation. A label between the verb and the causative verb helps in appropriate suffixation in target languages.



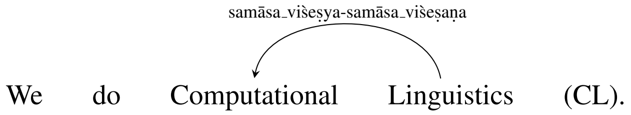
3. **kriyā-upasarga:**
   In a phrasal verb, this label connects verbs with their particles.



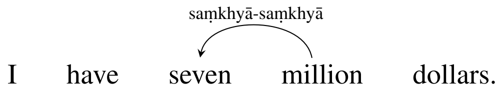




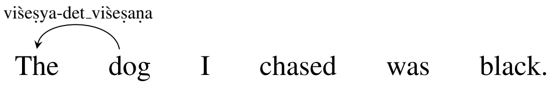
4. **samāsa_viŝeṣya-samāsa_viŝeṣaṇa:**
This label relates compound members, where a *samāsa_viŝeṣya* is the head noun in a compound and *samāsa_viŝeṣaṇa* is the modifier of the head noun.

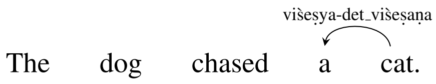

samāsa_viŝeṣya-samāsa_viŝeṣaṇa
We do Computational Linguistics (CL).

5. **saṃkhyā-saṃkhyā:**
In numerical constructions, this label connects the final number which is the head of the expression, with the preceding number.

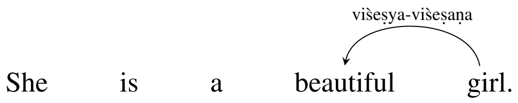


6. **viŝeṣya-det_viŝeṣaṇa:**
This label connects nouns with their determiners.





7. **viŝeṣya-viŝeṣaṇa:**
This label connects a *viŝeṣya* with *viŝeṣaṇa*. Also, this label is a mapping to the 'dep' label in Stanford dependency scheme. Stanford parser provides the label 'dep' when no specific relation label can be decided by the system. In general viŝeṣya reffers to a modified and viŝeṣaṇa to a modifier.



4.9 Conclusion

In this chapter, we talked about the way languages encode relational information through verbal and non-verbal affixes and positioning of the words in a sentence. We have shown how the concepts in Sanskrit grammar can help capture the surface level information in source language and how this information can be faithfully transferred to target language for relation extraction task.

We have shown that English encodes relational information through morphemes and position of the words such as pre-verbal and post-verbal positions and the position of members in compounds. We have made full use of such information in naming the relation labels. This makes the labels more informative and easy to understand.

The dependency tag schema presented in this chapter brings Stanford, LGP and CLEAR dependencies into one uniform notation which gives the system an ability to plug-in these parsers without any modifications in subsequent modules.

In future, the schema can be extended to extract *kāraka* level dependencies [8, 21] to represent deeper semantics of a sentence. The tag schema can also be used for building a dependency parser.