M A T E M A T I C K O - F Y Z I K Á L N Í F A K U L T A

P R A H A

TAMIL DEPENDENCY TREEBANK (TAMILTB) - 0.1
ANNOTATION MANUAL

LOGANATHAN RAMASAMY, ZDENĚK ŽABOKRTSKÝ

ÚFAL/CKL Technical Report
TR-2011-42

U N I V E R S I T A S C A R O L I N A P R A G E N S I S

Copies of ÚFAL/CKL Technical Reports can be ordered from:

Institute of Formal and Applied Linguistics (ÚFAL MFF UK)
Faculty of Mathematics and Physics, Charles University
Malostranské nám. 25, CZ-11800 Prague 1
Czech Republic

or can be obtained via the Web: http://ufal.mff.cuni.cz

Tamil Dependency Treebank (TamilTB) - 0.1 Annotation Manual

Loganathan Ramasamy and Zdeněk Žabokrtský

ÚFAL/CKL Technical Report
TR-2011-42

December 11, 2011

Contents

List of Tables

List of Figures

Abstract

Tamil Dependency Treebank (TamilTB) - 0.1 [Ramasamy and Žabokrtský, 2011] is an attempt to develop a syntactically annotated corpora for Tamil. TamilTB contains 600 sentences enriched with manual annotation of morphology and dependency syntax in the style of Prague Dependency Treebank. TamilTB has been created at the Institute of Formal and Applied Linguistics, Charles University in Prague. This report serves the purpose of how the annotation has been done at morphological level and syntactic level. Annotation scheme has been elaborately discussed with examples.

Chapter 1

Introduction

This chapter will briefly explain the Tamil Dependency Treebank (TamilTB) in general and various tasks involved in developing the TamilTB.

1.1 Background and Objectives

Treebank is an important resource in many Natural Language Processing (NLP) applications including parsers, language understanding, Machine Translation (MT) and so on. Treebanks are often manually developed and are available for only handful of languages such as English, German and Czech. Most of the world's other languages (irrespective of contemporariness and the number of speakers) do not have treebanks due to various reasons: (i) high development cost (ii) long development time (iii) lack of expertise to name a few. In this project, our main aim is develop a dependency treebank for Tamil language similar to Prague Dependency Treebank (PDT) annotation style.

We briefly introduce the work carried out on the subject prior to this work. There is an active research on dependency parsing [Bharati et al., 2009], [Nivre, 2009] and developing annotated treebanks for other Indian languages such as Hindi and Telugu. One such effort is, developing a large scale dependency treebank [Begum et al., 2008] (aimed at 1 million words) for Telugu, as of now the development for which stands [Vempaty et al., 2010] at around 1500 annotated sentences. For Tamil, previous works which utilised Tamil dependency treebanks are: [Dhanalakshmi et al., 2010] which developed dependency treebank (around 25000 words) as part of the grammar teaching tools, [Selvam et al., 2009] which developed small dependency corpora (5000 words) as part of the parser development. Other works such as [Janarthanam et al., 2007] focused on parsing the Tamil sentences. The current work differs from previous works with respect to the following objectives.

The main objectives of this project include,

- Annotate data at word level and syntactic level
- In each level of annotation, try for maximum level of linguistic representation
- Building large annotated corpora using automatic annotation process

1.2 Data

The data used for the TamilTB annotation comes from the news domain. We decided to use the news data for two reasons: (i) huge amount of data is available in digital format and can be easily downloadable and (ii) the news data can be considered as representative of written Tamil. At present, the data for the annotation comes from www.dinamani.com, and we downloaded pages randomly covering various news topics.

1.3 Text Preprocessing

Before the actual annotation takes place, the raw text data is preprocessed in three steps in sequential order,

No	Description	Value
1	Source	www.dinamani.com
2	Source transliterated	Yes
3	Number of words	9581
4	Number of sentences	600
5	Morphological annotation (sen)	600
6	Syntactic annotation (sen)	600
7	Tectogrammatical annotation	–

Table 1.1: The data for annotation

- Transliteration
- Sentence segmentation
- Tokenization

Each preprocessing step is explained in the following subsections.

1.3.1 Transliteration

The UTF-8 encoded Tamil raw text was transliterated to Latin for ease of processing during all levels of annotation. The raw UTF-8 encoded text can be obtained by applying reverse transliteration to the Latin-transcribed text. The transliteration scheme for Tamil script is given in the Figure 1.1. The Figure shows the transliteration for vowels, consonants, Sanskrit characters and an example transliterated sequence of consonant ('k')-vowel combination. Tamil has separate character representation for each consonant-vowel combination.[1]

Tamil Vowels	அ ஆ இ ஈ உ ஊ எ ஏ ஐ ஒ ஓ ஔ ஃ
Transliteration	a A i I u U e E ai o O au q
Tamil Consonants	க் ங் ச் ஞ் ட் ண் த் ந் ப் ம் ய் ர் ல் வ் ழ் ள் ற் ன்
Transliteration	k ng c nj t N T w p m y r l v z L R n
Sanskrit Characters	ஷ் ஹ் ஜ் ஸ்ரீ ஸ்
Transliteration	sh h j sri S
'k' + vowel combination	க கா கி கீ கு கூ கெ கே கை கொ கோ கௌ
Transliteration	ka kA ki kI ku kU ke kE kai ko kO kau

Figure 1.1: Transliteration scheme

Tamil is a phonetic language and the transliteration scheme is designed to match the Tamil sounds as much as possible. In this documentation, whereever possible, both Tamil script and transliterated form are used in examples. In few places, only transliterated format is used due to difficulties (mainly rendering problems) in embedding Tamil scripts.

1.3.2 Sentence Segmentation

Before the annotation takes place, the raw corpus downloaded from the source is sentence segmented automatically. This step can be performed before or after the transliteration. Like English, Tamil can also be ambiguous at various places that may look like sentence boundaries. But in reality they may not be sentence boundaries. Those ambiguous sentence boundaries (such as dots at decimal numbers, initials in names and dates) are detected through heuristics and the sentences are segmented only at appropriate places. As an example, in Tamil, the proper names are written using (*'surname name'*) format. But in usage the format is shortened to (**'initial.** *name*') where a dot is placed between *initial* and *name*. *Initial* is the first letter of a *surname*. Thus, the fullname *'Palaniappan Chidambaram'* in English will

[1] For full list of transliteration map, please refer this URL: https://ufal.mff.cuni.cz/~ramasamy/tamiltb/0.1/utf8_to_latin_map.txt (change the encoding of the browser to UTF-8 to view the contents properly)

be written as *'P. Chidambaram'* & *'pa. ciTambaram'* in English and Tamil respectively. The heuristics for this problem is straightforward. We listed down all possible Tamil characters (letters) and whenever a sentence termination symbol (*dot*) occurs after an initial will not be treated as a sentence boundary.

1.3.3 Tokenization

Tokenization is one of the important module that helps the annotation task. This module splits the sentence into words. Tamil uses spaces to mark word boundaries. But yet, a lot of Tamil wordforms are agglutinative in nature, meaning they glue together atleast two words (in majority of cases). Those cases can be identified as determiners+nouns, nouns+postpositions, verbs+particles, nouns+particles and etc. Except the first pattern (determiners+nouns), in all other cases, the second part of the wordforms are restricted and can be listed. So it is possible to split certain Tamil agglutinative wordforms into separate tokens. Certain particles (also called clitics) such as um/also, O/or are not treated as separate tokens in Tamil. But for the purpose of annotation we treat them as separate tokens. The same module will be used for tokenization when parsing the raw Tamil text.

Before splitting	Tamil	புதிய சட்டத்தின்படி , பாதுகாக்கப்பட்ட நினைவுச் சின்னத்திலிருந்து 1000 அடி வரை எந்த கட்டுமானமும் கட்ட அனுமதி இல்லை .
	Trans.	puTiya cattaTTin**pati** , pATukAkkap**patta** winaivuc cinnaTT**iliruwTu** 1000 ati varai ewTa kattumAnam**um** katta anumaTi illai .
After splitting	Tamil	புதிய சட்டத்தின் படி , பாதுகாக்கப் பட்ட நினைவுச் சின்னத்தி லிருந்து 1000 அடி வரை எந்த கட்டுமான மும் கட்ட அனுமதி இல்லை .
	Trans.	puTiya cattaTTin **pati** , pATukAkkap **patta** winaivuc cinnaTT **iliruwTu** 1000 ati varai ewTa kattumAnam **um** katta anumaTi illai .

Figure 1.2: Tokenization example

In the Figure 1.2, *pati* ('manner') and *iliruwTu* ('from') are postpositions, *patta* is an auxiliary verb and *um* is a clitic. This kind of agglutination is very prevalent in Tamil, and it would be useful to tagging process if we are able to reduce the vocabulary size by splitting the known combinations as separate tokens.

Clitics	**um**, *E, EyE, AvaTu*
Postpositions	*kUta, utan, pati, kuRiTTu,* **iliruwTu**, *anRu, uL, ARu, Tavira, pOTu, pOla, pinnar, pin, arukE, aRRa, inRi, illATa, mITu, kIz, mEl, munpE, otti, paRRi, paRRiya, pOnRa, mUlam, vaziyAka* etc.
Auxiliary Verbs	**patta,** *pattu, uLLa, pata, mAttATu, patuvArkaL, uLLAr, uLLanar, illai, iruwTAr, iruwTaTu, pattaTu, pattana, mutiyum, kUtATu, vENtum, kUtum, iruppin, uLLana, mutiyATu, patATu, koNtu, ceyTu* etc.
Particles	*Aka, Ana* and their spelling variants *Akac, AkaT, Akap, Akak*
Demonstrative pronouns	*ap, ac, ic, aw, iw* etc. as prefixes

Table 1.2: List of suffixes and words for tokenization

Some of the most commonly occurring (from the corpus) words and suffixes which participate in agglutination is given in the Table 1.2 above. Except demonstrative pronouns, all other words and suffixes are added after the stem. Among the categories in the Table 1.2 above. Clitics and Particles are the most participated in the agglutination. The tokenizer will make use of this list and try to separate these words from the original wordform. Even after the tokenization it would be possible to reconstruct the original sentence by making use of the attribute called `'no_space_after'`. The `'no_space_after'`

will be set to 1 if the following token is not separated from the current token. Whenever the splitting takes place this attribute will be set to 1 for the first token. For example, The `'no_space_after'` attribute for *pATukAkkap* will be 1. Whereas the `'no_space_after'` attribute for *um* will be 0. The splitting for the corpora has been done semi-automatically using some of the most commonly occuring combination from the above list and edited manually during the annotation process. At present, the tokenizer includes only few commonly occurring combinations from the Table 1.2 such as Clitics, Particles and very few postpositions.

We evaluated how much such combinations have been splitted from the original corpora. We found that 953 splits took place out of 9581 words. We simply did this by counting how many `'no_space_after'` attributes have been set to 1. We can say that almost 10% of the additional corpus size is due to splitting some wordforms into separate tokens.

1.4 Layers of Annotation

The annotation scheme used for TamilTB.v0.1 is similar to that of Prague Dependency Treebank 2.0 (PDT 2.0) [UFAL, 2006]. PDT 2.0 uses the notion *'layers'* to distinguish annotation at various levels (linguistic) such as word level and syntactic level. Precisely, PDT 2.0 is annotated at 3 levels or layers: (i) morphological layer (m-layer), (ii) analytical layer (a-layer) and (iii) tectogrammatical layer (t-layer). At present, TamilTB is annotated at only two layers: m-layer and a-layer.

Tamil	பண்பாட்டு அடையாளங்களைப் பாதுகாக்க தொல்பொருள் ஆய்வுத் துறை உருவாக்கப் பட்டு , தனிச் சட்டங்கள் இயற்றப் பட்டு உள்ளன .
Trans	paNpAttu ataiyALangkaLaip pATukAkka TolporuL AyvuT TuRai uruvAkkap pattu , Tanic cattangkaL iyaRRap pattu uLLana .
English	Having created Arachelogical Department, separate laws have been enacted to protect cultural symbols .

Figure 1.3: Tamil sentence example

In the the example shown in the Figure 1.3, the actual setence is given in Tamil script (indicated as Tamil:) in the 1st row, the transliterated (indicated as Tr:) version in the 2nd row, and the actual English translation in the 3rd row. The same format is used to illustrate sentence examples elsewhere in the document. There are 15 words in the Tamil sentence (including punctuations), each word will be treated as a node in each annotation layer. Please note that the term *node* will be used interchangeably with other terms *wordform* or *word* to represent a vertex in a-layer or m-layer.

Each node will have general attributes and layer specific attributes. For ex: a node in morphological layer will have attributes such as, 'lemma', 'form', 'tag' and `'no_space_after'` corresponding to lemma, wordfom, POS tag of a particular wordform and whether the following wordform is part of the current wordform . A node in analytical layer will have attributes such as dependency label ('afun') of the current node, whether the current node is an element in the coordination conjunction `('is_member')` etc. These attributes will be set automatically during parsing or editing attributes manually using TrEd. Also, the lower layer (m-layer) attributes are visible to upper layers (a-layer or t-layer).

Only transliterated version of the text will be used in all layers of annotation for the ease of processing. Examples in Tamil script are shown only for display purposes.

The following subsections briefly describe the annotation layers of TamilTB with an example.

1.4.1 Morphological Layer (m-layer)

The purpose of m-layer is to assign Parts of Speech (POS) tag or more refined morphological tag to each word in the sentence. This is accomplished by setting the *tag* attribute of the node (corresponds to word) to the POS or morphological tag. The *lemma* attribute will store the conceptual root or the word listed in dictionary as the lemma of the wordform. The Figure 1.4 illustrates m-layer annotation.

The Figure shows, there are three text values that are displayed at each node. The text at the top of the node is the *form* or the exact word which appeared in the text. The text at the middle (for ex: *paNpAttu*) of the node is the *lemma*, and the text at the bottom (for ex: NO - - 3SN - -) of the node is the morphological tag of the wordform. The length of each morphological tag is 9 characters and each character position will correspond to some feature of a wordform. The first 2 positions in the morpholgical

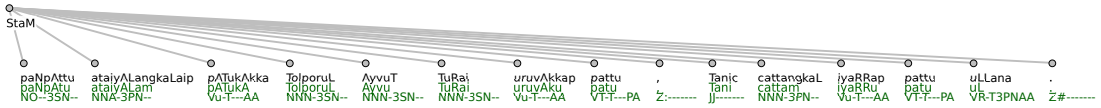


Figure 1.4: An example for morphological layer annotation

tag corresponds to main POS and refined POS. Both together will represent fine details of a wordform. Thus it is possible to train the POS tagger for a fine grained tagset or coarse grained tagset. This kind of tagging is known as positional tagging. Positional tagging is suitable for morphologically rich languages and has been successfully applied to languages such as Czech. The Chapter 2 gives a detailed description about positional tagging and the tagset used to perform annotation for TamilTB.

1.4.2 Analytical Layer (a-layer)

Analytical layer (a-layer) is used to annotate the sentence at syntactic level. There are two phases in a-layer annotation: (i) capture the dependency structure of a sentence in the form of tree and (ii) identify the relationship between words or nodes in the tree. From m-layer, we know that each wordform corresponds to a node in the tree but they are without their parents assigned. The dependency structure is captured by hanging the dependent nodes (words) under their governing nodes (words). Visually, dependent nodes will hang as children of their governing nodes. There will be one extra node called *technical root* to which the predicate node and the terminal node (end of the sentence) will be attached. The sole purpose of the *technical root* is to have some tree level attributes such as tree identifier. The Figure 1.5 illustrates the a-layer annotation of a sentence shown in Figure 1.3.

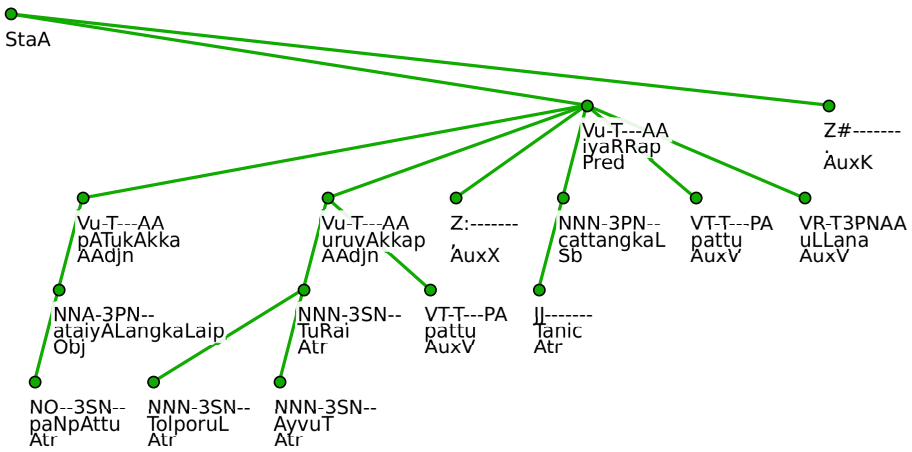


Figure 1.5: An example for analytical layer annotation

Edges between nodes indicate the relationship with which they are connected. In linguistic terms, it is called syntactic relation between governer and dependent. The relationship between two nodes are stored in the attribute called *afun*. In PDT style annotation, for technical reasons, the edges do not have *afun* attribute, instead the dependent nodes will store the *afun* attribute. For example, the *afun* value of the word *cattangkaL* ('laws') is `Sb` meaning Subject, is connected to the verb *iyaRRap* ('to enact').

The more detailed treatment of various syntactic relationships and a-layer annotation scheme is given in Chapter 3.

1.5 Obtaining Data

The annotated data is available in three formats,

1. TMT format - XML-based format used in the TectoMT system
2. CoNLL format - tabular eparated format in the CoNLL shared task style
3. TnT style POS tagged format - tabular separated columns with word forms, POS tags, and lemmas.

The single package containing the data and the documentation can be downloaded from the following link,

http://ufal.mff.cuni.cz/~ramasamy/tamiltb/0.1/download.html

The package can be uncompressed using the following command,

```
[shell]$ tar -zxvf TamilTB.v0.1.tar.gz
```

1.5.1 Contents of TamilTB.v0.1.tar.gz

- `TamilTB.v0.1/` (top level directory)
  - `TamilTB.v0.1.tmt` (in TMT format)
  - `TamilTB.v0.1.conll` (in CoNLL format)
  - `TamilTB.v0.1.tt` (only POS tagged corpora in TnT style)
- `doc`
  - `index.html` (main page)
  - ...
  - ...
  - ...
- `README.txt`

1.5.2 Prerequisites

The annotated data in TMT format requires tree editor TrEd to be installed. TrEd can be downloaded from,

http://ufal.mff.cuni.cz/~pajas/tred/

After installing TrEd, you need to install TMT plugin to browse the treebank data in the tmt file. The plugin can be installed by following,

- Start `TrEd`
- In menu, `[Setup -> Manage Extensions -> TMT files support]`
- Then treebank can be browsed by opening TamilTB.v0.1.tmt in TrEd

1.5.3 TectoMT Development Platform

The whole annotation work including preprocessing tasks (sentence segmentation, tokenization), manual annotation using tree editor and simple rule based parsing to bootstrap annotation have been implemented in TectoMT framework [Žabokrtský et al., 2008]. TectoMT is a highly modular NLP (Natural Language Processing) software system implemented in Perl programming language under Linux. It can be used for implementing various NLP tasks such as Machine Translation, POS tagging, parsing etc. in a modular way. For more about TectoMT, please refer the following URL,

http://ufal.mff.cuni.cz/tectomt/

1.5.4 Contact Authors

We welcome comments and suggestions for improvement of our current release. Please contact us for any suggestions or trouble in working the data.

Author Name	Email	Homepage
Loganathan Ramasamy	ramasamy@ufal.mff.cuni.cz	http://ufal.mff.cuni.cz/~ramasamy/
Zdeněk Žabokrtský	zabokrtsky@ufal.mff.cuni.cz	http://ufal.mff.cuni.cz/~zabokrtsky/

1.5.5 License

Chapter 2

Morphological Annotation

This chapter gives a detailed description of annotation at word level. The annotation at this layer is roughly equivalent to Part of Speech (POS) tagging. The chapter begins with the introduction of positional tag, a format for tagging wordforms. Then the chapter introduces positional tagset for tagging Tamil data with examples.

2.1 Positional Tagging

For m-layer annotation (aka POS tagging in general), we chose to annotate separate word tokens with morphological features in addition to single main POS. Having morphological features would be ideal and necessary for morphologically rich languages such as Tamil. Just by knowing those features it may be possible to identify certain syntactic phenomena (such as case markers can identify syntactic relations). So to include morphological information (such as case, person, number and etc.) in addition to main POS, we decided to adopt the positional tagging scheme which has been successfully applied for the Czech language.

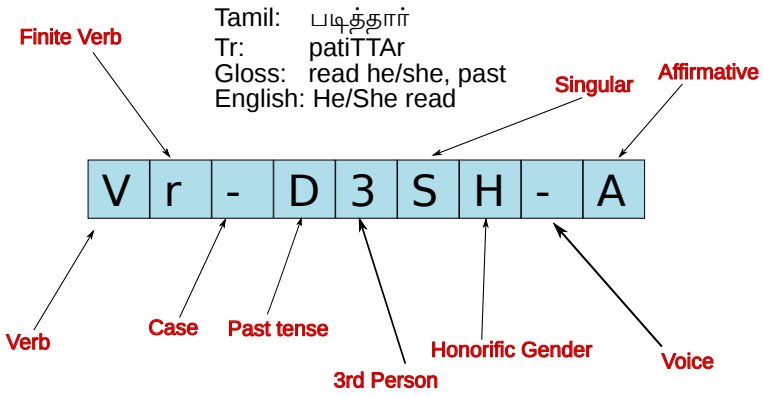


Figure 2.1: Positional tag

In positional tagging, each token (word) is tagged with a fixed length string. Each position or character in the tag represents a particular morphological feature of the token. The first position or character of the tag indicates the broad word category (such as noun, verb etc) to which the token belongs and the 2nd position indicates the detailed POS (for ex, what kind of verb? whether finite or non finite). The original Czech positional tagset includes 15 positions, and we designed a 9 positional tagset for Tamil. The Figure 2.1 illustrates the tagging of a Tamil word using positional tagging scheme.

The Figure 2.2 illustrates our positional tagset. The table in Figure 2.2(a) describes what each position occupies and the number of possible values they can take. The table in Figure 2.2(b) enumerates the possible values for *main POS* (first position). The tagset has been inspired from [Lehmann, 1989]. Our tagset includes separate entries for pronouns, numeral, interjections, particles and punctuations.

Position	Feature	#Possible Values
1	POS	14
2	Sub POS	42
3	Case	10
4	Tense	05
5	Person	04
6	Number	03
7	Gender	06
8	Voice	02
9	Negation	02

(a) Each position & num of possible values

Value	Description
A	Adverbs
C	Conjunctions
D	Determiners
I	Interjections
J	Adjectives
N	Nouns
P	Postpositions
Q	Quantifiers
R	Pronouns
T	Particles
U	Numerals
V	Verbs
X	Unknown
Z	Punctuations

(b) Main POS tags

Figure 2.2: Positional tagset

2.1.1 Position 2 - Sub POS

The *sub POS* corresponds to more finer version of *main POS* i.e. it can give some more information about the *main POS*. For ex, the *main POS* may just indicate that the wordform belongs to verb, the *sub POS* will further indicate that whether the verb is finite or non-finite. The *main POS* and *sub POS* together indicate the finer version of a part of speech. The other positions will further describe other details such as *tense, case, person, gender* and etc.

General note:

- For the tag of a given wordform, very rarely the entire positions of the tag is used. The value **'-'** has been used wherever the position is not applicable.
- For ex: In the tag *NNN-3SN-* -, the positions *4,8 and 9* are marked with *'-'* since they are not applicable for noun tag.

Table 2.1: SubPOS

POS	Sub POS	Description	Example	Tag	Comments
A	A	Adverbs, general	*maRupatiyum*	AA- - - - - - -	*maRupatiyum* ('again')
			vEkam Aka	NNN-3SN- -, AA- - - - - - -	*vEkam* ('fast, quick'). Most adverbs in Tamil can be formed by adding adverbial suffix *Aka* to the nouns. For example, *vEkamAka* ('quickly') can be formed by adding adverbial suffix *Aka* to the noun *vEkam* ('quick'). Though English doesn't split the adverb 'quickly' into 'quick' and 'ly', Tamil tokenizer splits those words with *Aka* suffix, because the suffix has other occurrences too where it appears as a separate word. Thus, only the adverbial suffix will receive the AA- - - - - - - tag and the noun will receive the noun tag.
C	C	Conjunctions	*maRRum*	CC- - - - - - -	*maRRum* ('and')
			allaTu	CC- - - - - - -	*allaTu* ('or')
D	D	Determiners	*iwTa*	DD- - - - - - -	*iwTa* ('this')
			awTa	DD- - - - - - -	*awTa* ('that')
			ewTa	DD- - - - - - -	*ewTa* ('which thing')
I	I	Interjections	*AhA*	II- - - - - - -	*AhA* ('oh')
J			*azak Ana*	NNN-3SN- -, JJ- - - - - - -	*azaku* ('beauty'). Adjectives can also be formed by adding adjectival suffix *Ana* to nouns. In that case, the nouns will receive the noun tag and the adjectival suffix will receive the adjective tag JJ- - - - - - -
	d	Adjective participle, adectivalized verbs	*OtukiRa*	Jd-P- - - - A	*OtukiRa* ('... which/who is running)
			Otiya	Jd-D- - - - A	*Otiya* ('... which/who ran')
			Otum	Jd-F- - - - A	*Otum* ('... which will run')
N	N	Common nouns	*kARRu*	NNN-3SN- -	*kARRu* ('air')
			kARRai	NNA-3SN- -	Accusative
			kARRAl	NNI-3SN- -	Instrumental
	E	Proper nouns	*kautamA*	NEN-3SH- -	*kautamA - 'Gautama'*
			pirAk	NEN-3SH- -	*pirAk - 'Prague'*

– continued on next page

Table 2.1 – continued from previous page

POS	Sub POS	Description	Example	Tag	Comments
	P	Participial nouns	*vAzwTavarkaL*	NPN-3PA- -	*vAzwTavarkaL - 'those who have lived'*
	O	Oblique nouns	*TirumaNa*	NO- - 3SN- -	When two nouns co occur the first noun will undergo certain change depending on the noun ending. They will act as a modifier to the second noun. Though they act as modifiers they will not be tagged as adjectives. They are tagged as oblique nouns. For ex: the noun *TirumaNam/'wedding'* becomes *TirumaNa/'wedding'* when it occurs as modifier to another noun say *'invitation'*. So it becomes, *TirumaNa azaippiTaz/'wedding invitation'*
P	P	Postpositions	*mItu*	PP- - - - - - -	*mITu - 'on'*
			itamiruwTu	PP- - - - - - -	*itamiruwTu - 'from'*
					Other examples: pati - 'according', pinnar - 'later, after', mUlamAka - 'through'
Q	Q	quantifiers	*aTikam, konjcam, mikavum, muzu*	QQ - - - - - - -	('a lot'), *konjcam* ('a bit') etc.
R	p	Personal pronouns	*avar*	RpN-3SH- -	*avar - 'he or she honorific*
			iTu	RpN-3SN- -	*iTu - 'this'*
			wAm	RpN-1PA- -	*wAm - 'we'*
	h	Reflexive pronouns	*TanaTu*	RhG-3SA- -	*TanaTu - "oneself's"*
			Tannai	RhA-3SA- -	*Tannai – 'about oneself'*
	B	General referential pronouns	*yArum*	RBN-3SA- -	*yArum - 'anyone'*
			eTuvum	RBN-3SN- -	*eTuvum - 'any thing'*
			yArukkum	RBD-3SA- -	*yArukkum - 'to anyone'*
	F	Specific indefinite referential pronouns	*yArO*	RFN-3SA- -	*yArO - 'someone'*
			eTuvO	RFN-3SN- -	*eTuvO - 'something'*
	I	Inerrogative pronouns	*yAr*	RiN-3SA- -	*yAr - 'who'*

– continued on next page

Table 2.1 – continued from previous page

POS	Sub POS	Description	Example	Tag	Comments
			eTu	RiN-3SN- -	*eTu - 'which thing'*
	G	Non specific indefinite pronouns	*yArAvaTu*	RGN-3SA- -	*yArAvaTu - 'at least someone'*
			eTAvaTu	RGN-3SN- -	*eTAvaTu - 'at least something'*
T	b	Comparative particle	*kAttilum, vita*	Tb - - - - - - -	*kAttilum, vita* ('than')
	d	Adjectival participle, adectivalized verbs	*enkiRa, enRa*	Td-P- - - - A	*enkiRa* ('the so called')
	e	Interrogative particle	*A*	Te - - - - - - -	*A* ('is it ...?')
	g	Adverb and Adjectival suffix	*Aka, Ana*	Tg - - - - - - -	*Aka, Ana*
	k	Intensifier particle	*E, EyE, TAn*	Tk - - - - - - -	*E, TAn* ('indeed')
	l	Clitic	*AvaTu*	Tl - - - - - - -	*AvaTu* ('at least') or it can represent indefiniteness in combination with pronouns.
	m	Clitic (*limit*)	*mattum*	Tm - - - - - - -	*mattum* ('limit')
	n	Complementizing nouns	*pOTu*	Tn- - - - - - -	*pOTu - 'when, during, while'*
			utan	Tn- - - - - - -	*utan - 'as soon as'* **Other examples:** *piRaku - 'after', mun - 'before', varai - 'till, as long as', mATiri - 'manner', pati - 'manner'*
	o	Particle of doubt	*rAmanO*	To - - - - - - -	*rAmanO* ('whether it is Raman ...')
	q	Emphatic particle	*E, EyE* and *TAn*	Tq - - - - - - -	*E, EyE* and *TAn*. For ex: *ennaiyE* ('just me')
	Q	Complementizer	*enpaTu*	TQ - - - - - - -	*enpaTu*
	s	Concessive particle	*Otiyum*	Ts - - - - - - -	*um* ('although, even if'). For ex: *Otiyum* ('although ran')
	S	Immediacy particle	*vawTaTum*	TS - - - - - - -	*um* ('as soon as'). For ex: *vawTaTum* ('as soon as came').

– continued on next page

Table 2.1 – continued from previous page

POS	Sub POS	Description	Example	Tag	Comments
	t	Complementizer in verbal participle	*ena, enRu*	Tt - T - - - - A	*ena, enRu.*
	v	inclusive particle	*um, kUta*	Tv - - - - - - -	*rAmanum* ('also Ram')
	w	complementizer in conditional form	*enRAl*	Tw - T - - - - A	*enRAl*
U	x	cardinals	*onRu, iraNtu*	Ux - - - - - - -	*onRu, iraNtu* ('one, two').
	y	ordinals	*onRAm, iraNtAm*	Uy - - - - - - -	*onRAm, iraNtAm* ('first, second')
	=	digits	*10,20*	U = - - - - - - -	numbers using digits.
V	j	imperative verb (lexical)	*Otu, uTavu*	Vj - T2PAAA	*irungkaL* ('you/[polite] wait')
	r	finite verb (lexical)	*OtukiRAn, uTavukiRAL*	Vr - P3SMAA	*OtukiRAn* ('he runs/running')
	t	verbal participle (lexical)	*Oti, vawTu*	Vt - T - - - AA	*Oti* ('having run')
	u	infinitive (lexical)	*Ota, uTava*	Vu - T - - - AA	*Ota, uTava* ('to run, to help')
	w	conditional verb (lexical)	*OtinAl, vawTAl*	Vw - T - - - AA	*OtinAl, vawTAl* ('if runs/ran, if comes/came')
	z	verbal nouns (lexical)	*pitippaTu*	VzNF3SNAA	*pitippaTu* ('capturing')
	R	finite verb (auxiliary)	*iruwTAr*	VR - D3SHAA	*iruwTAr*
	T	verbal participle (auxiliary)	*koNtu*	VT - T - - - AA	*koNtu*
	U	infinitive (auxiliary)	*patta*	VU - T - - - AA	*patta*
	W	conditional (auxiliary)	*pattAl*	VW - T - - - AA	*pattAl*

– continued on next page

Table 2.1 – continued from previous page

POS	Sub POS	Description	Example	Tag	Comments
	Z	verbal nouns (auxiliary)	*iruppaTu*	VZNF3SNAA	*iruppaTu*
Z	# :	terminal symbol other symbols	.(dot) *, , ? , -, (,), !* etc.	Z# - - - - - - - Z: - - - - - - -	sentence terminating symbol (usually *period*) all other symbols other than sentence termination symbol

2.1.2 Position 3 - Case

Tamil case occupies third position in the positional tag. The Table 2.2 lists Tamil *case markers* and the corresponding values for the 3^{rd} position of the positional tag. There is a disagreement over the number of *case markers* in Tamil. [Lehmann, 1989] defines 9 cases. Some of the *case markers* can be considered as a *bound postpositions*. For that reason, we have not included the cases such as *ablative* and *benefactive* which are treated as postpositions.

#	Value	Description	Example	Tag
1	A	Accusative	*katciyai* ('party')	NNA - 3SN - -
2	D	Dative	*vIttukku* ('to/for the house')	NND - 3SN - -
3	I	Instrumental	*muyaRciyAl* ('by the efforts')	NNI - 3SN - -
4	G	Genitive	*aracin* ('government's')	NNG - 3SN - -
5	L	Locative	*pOril* ('in the war')	NNL - 3SN - -
6	N	Nominative	*ANtu* ('year')	NNN - 3SN - -
7	S	Sociative	*TuNaiyOtu* ('with the help')	NNS - 3SN - -

Table 2.2: Tamil case

2.1.3 Position 4 - Tense

#	Value	Description	Example	Tag
1	D	past	*kattinAr* ('built he')	Vr - D3SHAA
2	F	future	*uTavum* ('it will help')	Vr - F3SNAA
3	P	present	*celkiRAr* ('he is going')	Vr - P3SHAA
4	T	tenseless	*illai* ('exist not')	Vr - T3PNAA

Table 2.3: Tense

Note: Tenseless verbs

1. Tenseless verbs are tagged with **T**.
2. Tenseless verbs are unmarked in the wordform for tense.
3. Examples: *infinitives, verbal participles, imperatives* and some exceptional verbs.

2.1.4 Position 5 - Person

#	Value	Description	Example	Tag
1	1	1^{st} person	*mERkoNtEn* ('I undertook')	Vr - D1SAAA
2	2	2^{nd} person	*anjcukiRIrkaL* ('you fear')	Vr - P2PAAA
3	3	3^{rd} person	*vivATikkum* ('it will discuss')	Vr - F3SNAA
4	X	*unused*	*unused*	*unused*

Table 2.4: Person

The value *X* is reserved for future purposes. At present, *X* has not been used, rather the tag - is used the category is not applicable.

2.1.5 Position 6 - Number

#	Value	Description	Example	Tag
1	P	plural	*vivarangkaL* ('details')	NNN - 3PN - -
2	S	singular	*nyUSilAwTu* ('New Zealand')	NEN - 3SN - -
3	X	*unused*	*unused*	*unused*

Table 2.5: Number

2.1.6 Position 7 - Gender

#	Value	Description	Example	Tag
1	F	feminine	*varuvAL* ('she will come')	Vr - F3SFAA
2	M	masculine	*Atavanin* ('man's')	NNG - 3SM –
3	N	neuter	*etuTTaTu* ('it took')	Vr- D3SNAA
4	H	honorific (both masc. and fem.)	*avar* ('he/she [polite]')	RpN - 3SH - -
5	A	animate (humans)	*yAr* ('who?')	RiN - 3SA - -
6	I	inanimate (non humans)	*unused*	*unused*
7	X	*unused*	*unused*	*unused*

Table 2.6: Gender

2.1.7 Position 8 - Voice

#	Value	Description	Example	Tag
1	A	active	*etuTTaTu* ('it took')	Vr- D3SNAA
2	P	passive	*patukiRaTu* ('being [*verb*]...')	VR - P3SNPA

Table 2.7: Voice

2.1.8 Position 8 - Negation

#	Value	Description	Example	Tag
1	A	affirmative	*pinpaRRa* ('to follow')	Vu - T - - - AA
2	N	negation	*mutiyATu* ('cannot')	VR - T3SN -A

Table 2.8: Negation

2.2 Annotation of Pronouns

Tamil pronouns are one of the closed but in combination with clitics produce various derived pronouns. In this section, we list all possible pronouns (in their combination) with their tags. More information about Tamil pronouns can be obtained from [Lehmann, 1989]. The listing of tags for all possible pronouns will be useful in annotation task.

2.2.1 Singular Referential Definite (Personal) Pronouns

#	Person/Number	Pronoun	Tag
1	1^{st}/singular	*wAn* ('I')	RpN - 1SA - -
2	1^{st}/plural	*wAm* ('we, exclusive')	RpN - 1PA - -
3	"	*wAngkaL* ('we, inclusive')	RpN - 1PA - -
4	2^{nd}/singular	*wI* ('you')	RpN - 2SA - -
5	"	*wIngkaL* ('you, honorific, singular')	RpN - 2SH - -
6	2^{nd}/plural	*wIngkaL* ('you, plural')	RpN - 2PA - -
7	3^{rd}/singular	*avan* ('that one - he')	RpN - 3SM - -
8	"	*ivan* ('this one - he')	RpN - 3SM - -
9	"	*avaL* ('that one - she')	RpN - 3SF - -
10	"	*ivaL* ('this one - she')	RpN - 3SF - -
11	"	*aTu* ('that one - it')	RpN - 3SN - -
12	"	*iTu* ('this one - it')	RpN - 3SN - -
13	"	*avar* ('that one - he/she hon.')	RpN - 3SH - -
14	"	*ivar* ('this one - he/she hon.')	RpN - 3SH - -
15	3^{rd}/plural	*avai/avaikaL* ('those ones')	RpN - 3PN - -
16	"	*ivai/ivaikaL* ('these ones')	RpN - 3PN - -
17	"	*avarkaL* ('those people')	RpN - 3PA - -
18	"	*ivarkaL* ('these people')	RpN - 3PA - -

Table 2.9: Personal pronouns

2.2.2 Non Referential (Interrogative) Pronouns

#	Pronoun	Tag
1	*yAr* ('who')	RiN - 3SH - -
2	*enna* ('what')	RiN - 3SN - -
3	*evan* ('which male person')	RiN - 3SM - -
4	*evaL* ('which female person')	RiN - 3SF - -
5	*eTu* ('which thing')	RiN - 3SN - -
6	*evar* ('which male/female person')	RiN - 3SH - -
7	*evarkaL* ('which persons')	RiN - 3PA - -
8	*evai(kaL)* ('which things')	RiN - 3PN - -

Table 2.10: Interrogative pronouns

2.2.3 General Referential Pronouns

In the case of general referential pronouns, the particle *-um* is added to interrogative pronouns which results in the addition of referential property to interrogatives. The resultant words will have a meaning of 'anyone, anybody or anything' in English. The Table 2.11 lists general referential pronouns.

Note

- Though the *inclusive* particle *-um* is added to pronouns, the tokenizer does not split the suffix from the pronoun unlike other situations.
- In most other cases, the particle *-um* is splitted from wordforms.

#	Pronoun	Tag
1	*yAr***um** (‘anyone’)	RBN - 3SH - -
3	*evan***um** (‘anyone, male person’)	RBN - 3SM - -
4	*evaL***um** (‘anyone female person’)	RBN - 3SF - -
5	*eTuv***um** (‘anything’)	RBN - 3SN - -
6	*evar***um** (‘anyone, male/female person’)	RBN - 3SH - -
7	*evarkaL***um** (‘any persons’)	RBN - 3PA - -
8	*evaikaL***um** (‘anything, plural’)	RBN - 3PN - -
8	*evaiy***um** (‘anything, plural’)	RBN - 3PN - -

Table 2.11: General referential pronouns

2.2.4 Specific Indefinite Referential Pronouns

#	Pronoun	Tag
1	*yAr***O** (‘someone’)	RFN - 3SH - -
3	*evan***O** (‘some male person’)	RFN - 3SM - -
4	*evaL***O** (‘some female person’)	RFN - 3SF - -
5	*eTuv***O** (‘something’)	RFN - 3SN - -
6	*evar***O** (‘some male/female person’)	RFN - 3SH - -
7	*evarkaL***O** (‘someone, plural’)	RFN - 3PA - -
8	*evaikaL***O** (‘something, plural’)	RFN - 3PN - -
8	*evaiy***O** (‘something, plural’)	RFN - 3PN - -

Table 2.12: Specific indefinite referential pronouns

2.2.5 Non-specific Indefinite Referential Pronouns

#	Pronoun	Tag
1	*yAr***AvaTu** (‘someone or other’)	RGN - 3SH - -
3	*evan***AvaTu** (‘some male person or other’)	RGN - 3SM - -
4	*evaL***AvaTu** (‘some female person or other’)	RGN - 3SF - -
5	*eT***AvaTu** (‘something or other’)	RGN - 3SN - -
6	*evar***AvaTu** (‘some male/female person or other’)	RGN - 3SH - -
7	*evarkaL***AvaTu** (‘someone, or other (plural)’)	RGN - 3PA - -
8	*evaikaL***AvaTu** (‘something, or other (plural)’)	RGN - 3PN - -
8	*evaiy***AvaTu** (‘something, other (plural)’)	RGN - 3PN - -

Table 2.13: Non-specific indefinite referential pronouns

Chapter 3

Syntactic Annotation

This section will give a detatiled description about how annotation takes place at the syntactic level. The syntactic annotation consists of two phases: (i) identifying the structure (dependency) of the sentence in the form of dependency tree and (ii) identifying the dependency relations and assigning those relations to edges in the dependency tree structure.

3.1 Identifying the Structure

The structure of the sentence is identified manually by attaching the dependent nodes to the governing nodes. In the sentential structure, the head of the sentence will be predicate, and the predicate will have arguments (noun phrases, adverbials) as their children. The objective of this step would be, identifying the predicate rooted structure and attaching to the techincal root (AuxS, defined below) of the tree. The end of the sentence will also be attached to the technical root. Once the structure is identified, all the edges have to be labeled with their relations. For technical reasons, the relation between the dependent and the governing node is stored as an a-layer attribute of the dependent node. The attribute is called *'afun'*. The following sections explains each dependency relation in detail.

3.2 Dependency Relations

According to PDT naming convention, dependency relations are also called as *analytical functions* or *afuns*. The documentation uses these names interchangeably.

#	Afun	Description	Comments
1	AAdjn	Adverbial Adjunct	Optional adverbs, optional PP phrases attaching to verb
2	AComp	Adverbial Complement	Obligatory adverbs, obligatory PP phrases attaching to verb
3	Apos	Apposition	Heads of the apposition clauses - clauses attaching to *enRa*
4	Atr	Attribute	Noun modifiers
5	AdjAtr	Adjectival participial	Adjectivalized verbs, or relative clauses
6	AuxA	Determiners	Demonstrative pronouns *iwTa* (‘this’), *awTa* (‘that’)
7	AuxC	Subordinating conjunctions	Subordinating Conjunctions - *enRu, ena, Aka*
8	AuxG	Symbols other than comma	-, ”, ’, $, rU., (,), [,] etc.
9	AuxK	Sentence termination symbols	:, . , ?
10	AuxP	Postpositions	*mITu* (‘on’), *paRRi* (‘about’), *kIz* (’under’)
11	AuxS	Technical root	Technical root
12	AuxV	Auxiliary verb	*uL, koNtu, iru* etc.
13	AuxX	Comma (not coordination)	,
14	AuxZ	Emphasis words or particles	*TAn* (emphasis), *um* (also, even), *E* (’even’)
15	CC	Part of a word	*kiLarwTu ezuwTu* - (’rising’). *kiLarwTu* will be labeled as *CC*
16	Comp	Complement other than attaching to verbs	Obligatory attachments to non verbs. Ex: “belongs to the batch of 1977”
17	Coord	Coordination node	*maRRum* (‘and’), *um*
18	Obj	Object (both direct and indirect)	usually nouns with accusative case
19	Pnom	Predicate nominal	Nominals that act as main verbs
20	Pred	Predicate	Main verb (usually finite) of a sentence.
21	Sb	Subject	Subject usually nominals

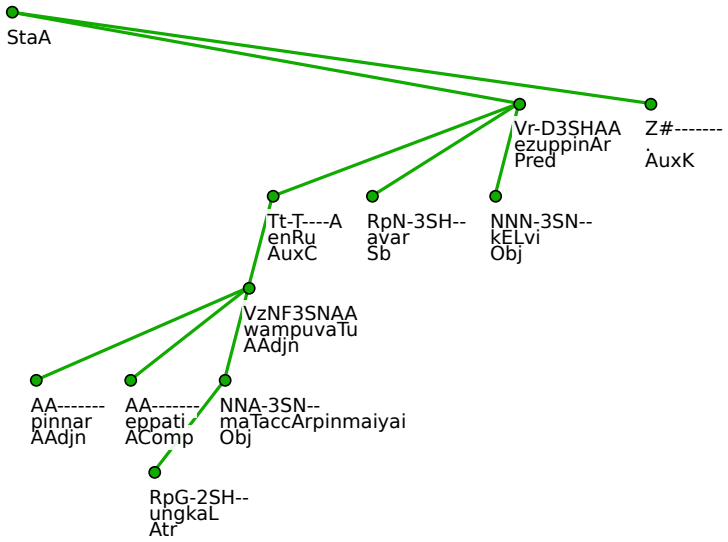
Table 3.1: Dependency relations *(afuns)*

3.3 Detailed Description of *afuns*

In this section, we describe each dependency relation in detail with example annotation of that relation. For the list of dependency relations or *afuns*, please refer the Table 3.1. Each dependency relation is explained with an analytical tree and the sentence it represents. The sentence is shown in Tamil script and it's transliteration, gloss and the actual English translation in that order. The word shown in bold will receive the dependency relation that is being explained.

3.3.1 Afun: *AAdjn*

The *AAdjn* relation is used to mark adverbial adjuncts. Adverbial adjuncts are optional adverbial phrases, prepositional phrases, clauses or simple adverbs modifying the verbs. Figures 3.1 & 3.3 show examples for *AAdjn* relation. In the Figure 3.1, the adverb *pinnar* ('later, after') has been labeled with *AAdjn* relation.



Tamil: பின்னர் எப்படி உங்கள் மதச்சார்பின்மையை நம்புவது என்று கேள்வி எழுப்பினார் .
Tr: **pinnar** eppati ungkaL maTaccArpinmaiyai wampuvaTu enRu kELvi ezuppinAr .
Gloss: **then** how your secularism to-believe - question raised-he .
English: "**Then** how to believe your secular credentials ", he raised a question .

Figure 3.1: *AAdjn:* Adverbial adjunct

In the Figure 3.3, the adverbial phrase *munniRuTTi* ('by putting forward ...') has been labeled with *AAdjn* relation. The *AAdjn* label is determined by whether excluding the adverbial adjunct affects the meaning of the sentence. If it does not (only provides extra information about the sentence), the head of the phrase is assigned *AAdjn* relation.

3.3.2 Afun: *AComp*

The *AComp* relation is used to mark the obligatory adverbials or adverbial complements in the structure. The context of occurrence of *AComp* relation is same as that of *AAdjn* relation. The only difference is that the *adverbial adjuncts (AAdjn)* are optional elements in the sentential structure whereas *adverbial complements (AComp)* are obligatory elements in the sentence structure. While doing annotation, this relationship is determined by whether the adverbial structure is required to complete the sentence. If the removal of the adverbial structure does not affect the sentence as a whole, then it is labeled as *AAdjn* otherwise it will be labeled as *AComp*.

- Our annotation treats the adverbials as *adverbial complements (AComp)* and *adverbial adjuncts (AAdjn)*. Whereas the orignal PDT treats the both under *adverbials (Adv)*.

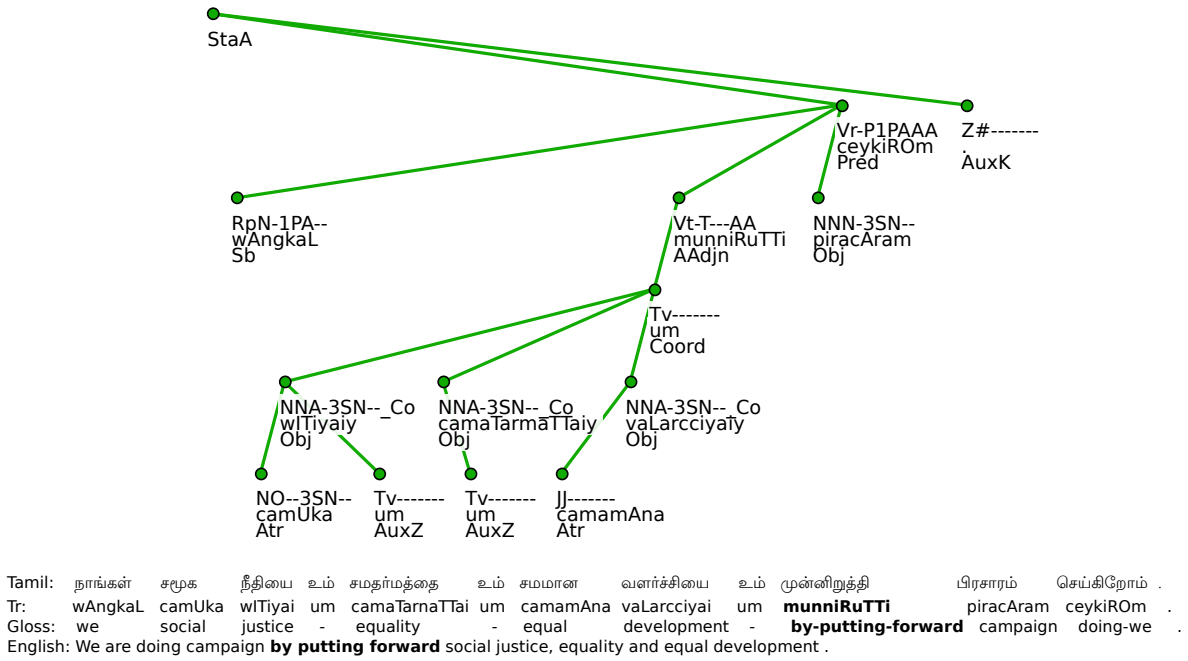


Figure 3.2: *AAdjn:* Adverbial adjunct

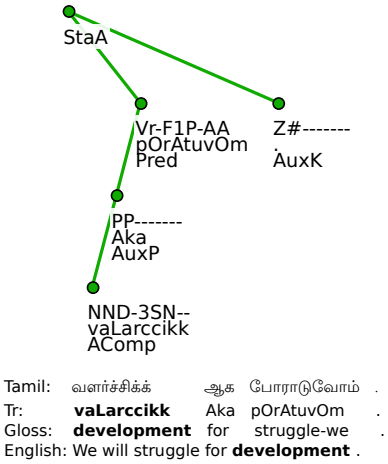


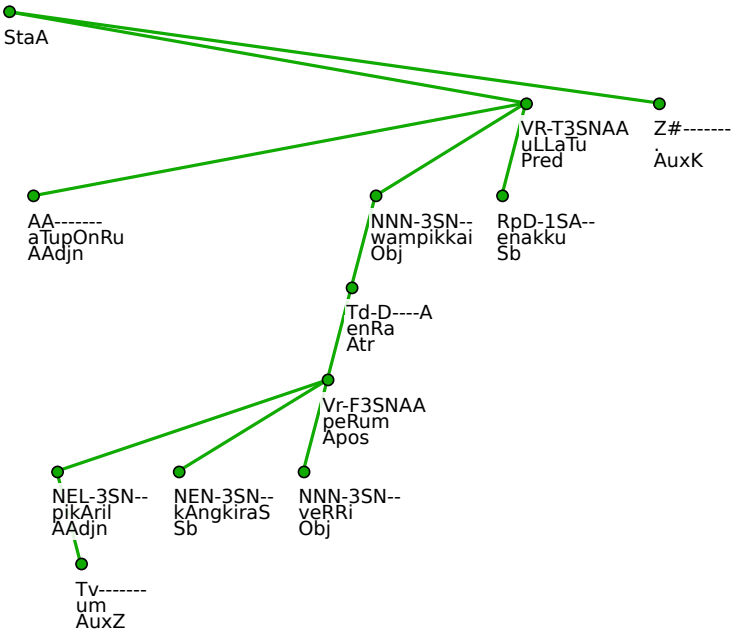
Figure 3.3: *AComp:* Adverbial complement

3.3.3 Afun: *Apos*

The adjectival clauses headed by *enRa* are appositional clauses. The entire finite clause will be attached to *enRa* which will act as a modifier to the following noun phrase. The clausal head which is attached to *enRa* will receive Apos label. The following Figure 3.4 illustrates the labeling of *Apos*.

3.3.4 Afun: *Atr*

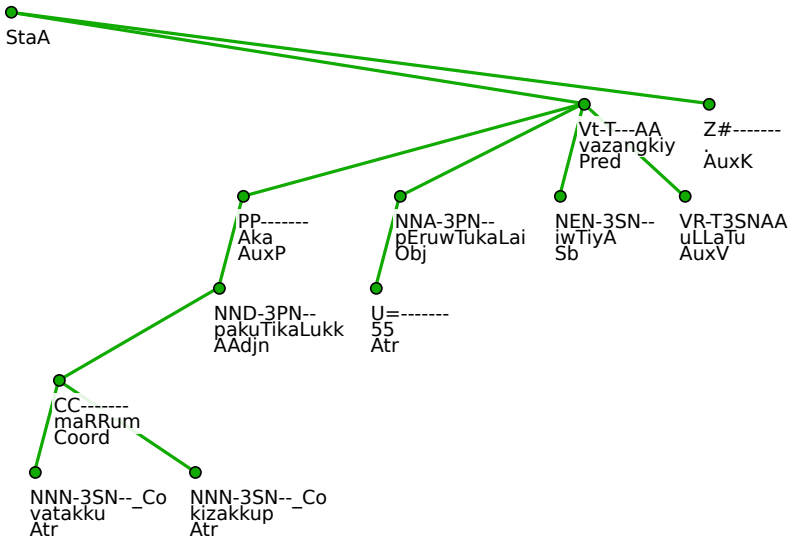
Attribute [UFAL, 2006] is a sentence member which depends on noun and closely determines its meaning. Original PDT annotation diferentiates “agreeing” and “non-agreeing” attribute. But, since Tamil does not have any agreement between nouns and their modifers, all noun modifers will receive the afun label *Atr*. The noun modifiers include nouns (except the head noun) in noun compounds, adjectives, numerals and adjectival participles. The Figure 3.5 shows the usage of afun *Atr*.



Tamil:	அதுபோன்று	பிகாரில்	உம்	காங்கிரஸ்	வெற்றி	பெறும்	என்ற	நம்பிக்கை	எனக்கு	உள்ளது	.
Tr:	aTupOnRu	pikAril	um	kAngkiraS	**veRRi**	peRum	enRa	wampikkai	enakku	uLLaTu	.
Gloss:	Like-that	in-Bihar	also	Congress	**victory**	gain	that	confidence	I	AUX-PERF	.

English: Like that, I have a confidence that in Bihar too Congress will gain **victory** .

Figure 3.4: *Apos:* Apposition



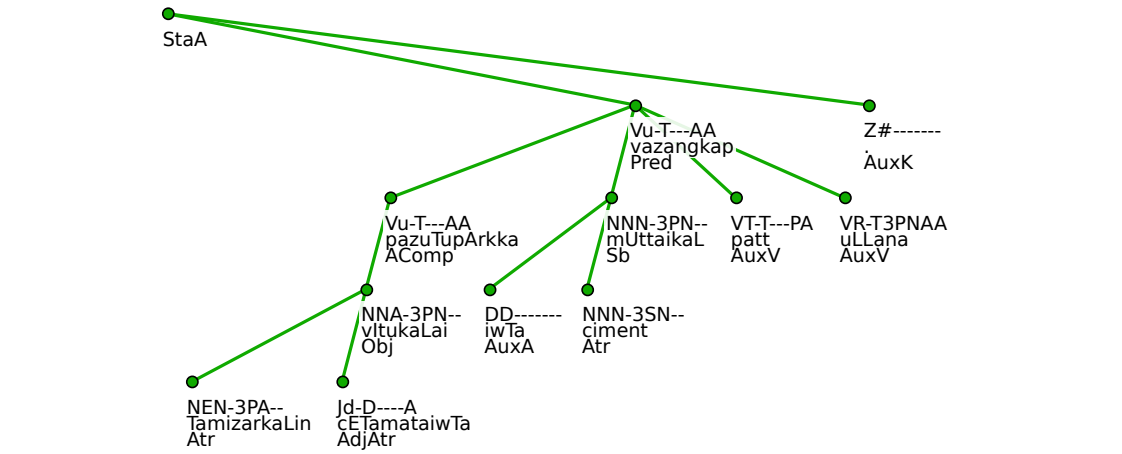
Tamil:	வடக்கு	மற்றும்	கிழக்குப்	பகுதிகளுக்க்	ஆக	55	பேருந்துகளை	இந்தியா	வழங்கிய்	உள்ளது	.
Tr:	**vatakku**	maRRum	**kizakkup**	pakuTikaLukk	Aka	**55**	pEruwTukaLai	iwTiyA	vazangkiy	uLLaTu	.
Gloss:	**north**	and	**east**	regions	for	**55**	buses	India	give	AUX-PERF	.

English: India has provided 55 buses to north and eastern regions .

Figure 3.5: *Atr:* Attribute

3.3.5 Afun: *AdjAtr*

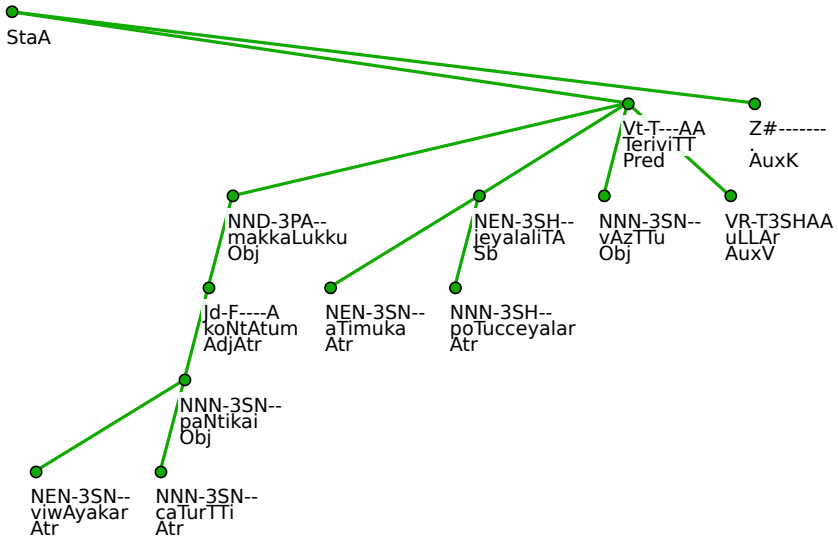
AdjAtr label is used to mark the adjectival clauses, adjectivalized verbs or adjectival participials. They are equivalent to -ing, -ed (singing girl, departed train) forms in English. Verbs in Tamil can be adjectivalized for all three tenses, and they take appropriate tags depending on the word form features. Figures 3.6 & 3.7 show examples for *AdjAtr* labeling.



Tamil:	தமிழர்களின்	சேதமடைந்த	வீடுகளை	பழுதுபார்க்க	இந்த	சிமெண்ட்	மூட்டைகள்	வழங்கப்	பட்ட்	உள்ளன	.
Tr:	TamizarkaLin	**cETamataiwTa**	vItukaLai	pazuTupArkka	iwTa	cimeNt	mUttaikaL	vazangkap	patt	uLLana	.
Gloss:	Tamil's	**ruined**	houses	to-repair	these	cement	-	to-provide	AUX-PASS	AUX-PERF	.

English: These cement stocks have been provided to repair the ruined houses of Tamils.

Figure 3.6: *AdjAtr:* Adjectival attribute



Tamil:	விநாயகர்	சதுர்த்தி	பண்டிகை	கொண்டாடும்	மக்களுக்கு	அதிமுக	பொதுச்செயலாளர்	ஜெயலலிதா	வாழ்த்து	தெரிவித்த்	உள்ளார்	.
Tr:	viwAyakar	caTurTTi	paNtikai	**koNtAtum**	makkaLukku	aTimuka	poTucceyalALar	jeyalaliTA	vAzTTu	TeriviTT	uLLAr	.
Gloss:	Vinayakar	Chaturthi	festival	**celebrating**	people-DAT	ADMK	general-secretary	Jeyalalitha	wishes	express	AUX-PERF	.

English: ADMK General Secretary Jeyalalitha expressed her wishes to people who are celebrating Vinayakar Chaturthi festival .

Figure 3.7: *AdjAtr:* Adjectival attribute

3.3.6 Afun: *AuxA*

Tamil has two demonstrative determiners corresponding to 'this' and 'that' in English. At present, the question word *ewTa* ('which') is also tagged as determiner. Sometimes the determiners occur as a prefix to the following noun or noun phrase in a contracted form. During the tokenization phase, these demonstrative suffixes will be separated from the noun or noun phrase, and will be considered as a separate token. The determiners are *iwTa* ('this'), *awTa* ('that') and *ewTa* ('which'), and the corresponding contracted determiners are *i, a* and *e*. Due to orthorgraphic rules, the first letter of the following noun phrase will be added to the contracted form. The Figure 3.8 illustrates the usage of all determiners.

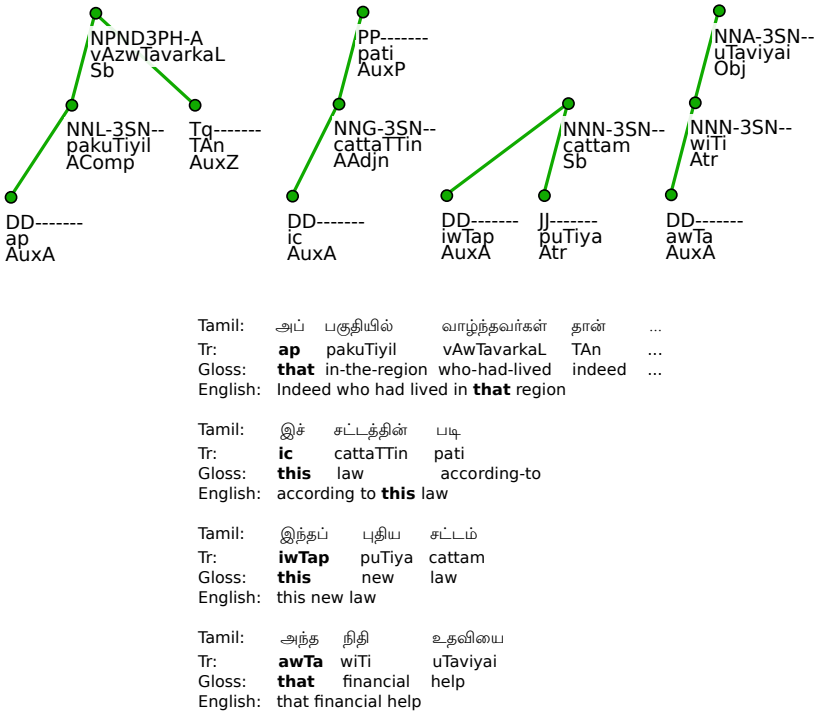


Figure 3.8: *AuxA:* Determiners

3.3.7 Afun: *AuxC*

In Tamil, embedding or adjoining of clauses are performed either by morphologically marking the clause or by using separate words. When separate words are used, they function similar to that of subordinating conjunction words in other languages such as English. These separate words are called complementizers in Tamil. Complementizers can be verbs, nouns or postpositions after nominalized clauses. There are three complementizing verbs - *en* ('say'), *pOl* ('seem') and *Aku* ('become'). They have grammatical function during embedding of clauses, otherwise they retain their lexical meaings. The following list provides some of the noun complementizers - *pOTu* ('time'), *mun* ('before'), *piRaku* ('after'), *utan* ('immediacy'), *varai* ('as long as') and etc. The postpositions can also be interpreted as subordinating conjunction words when they are preceded by nominalized clauses. Refer [Lehmann, 1989] for detailed treatment of how complementizers work in Tamil.

3.3.8 Afun: *AuxG*

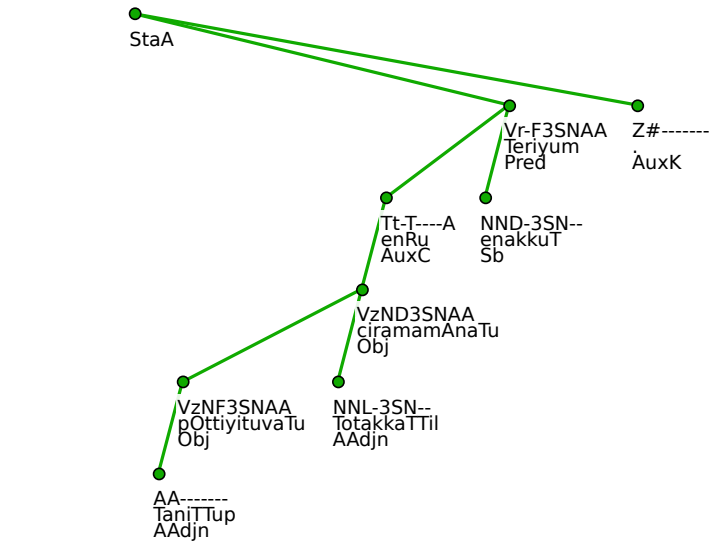
The symbols other than sentence boundary and comma are labeled with *AuxG* afun. The *AuxG* symbols include pairs of symbols such as *(, }, [, “, ‘* and other symbols such as !, , #,$ and etc. The Figure 3.11 shows an example for *AuxG*.

3.3.9 Afun: *AuxK*

The *AuxK* afun is assigned to sentence termination symbols. The symbols *., ?* and *:* are considered as sentence terminals and they are expected at the end of a sentence.

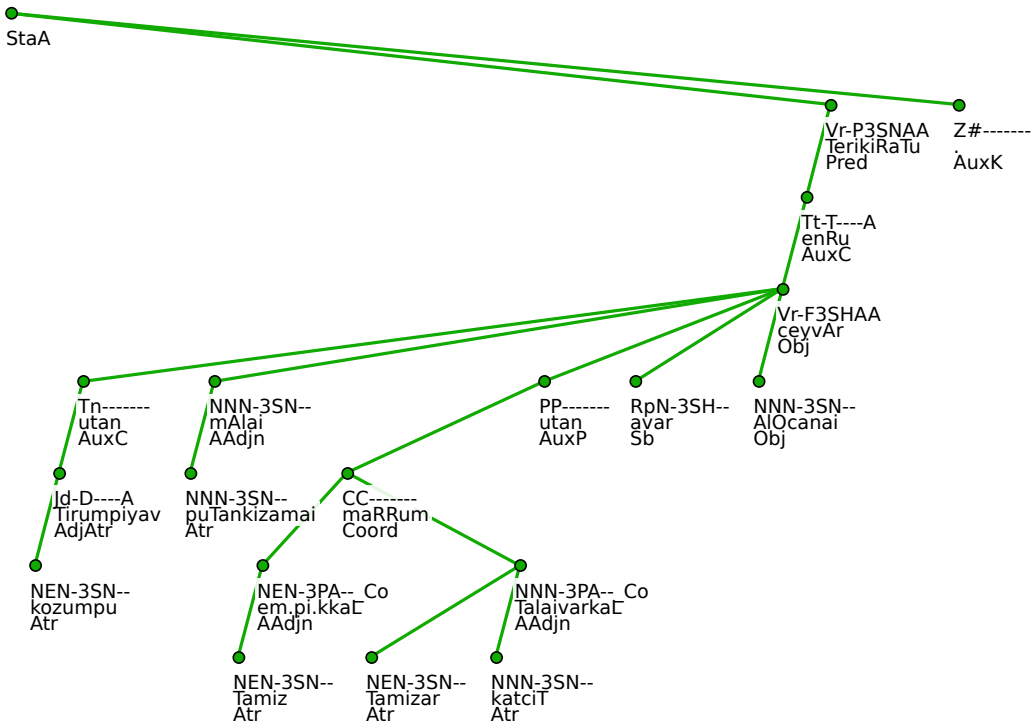
3.3.10 Afun: *AuxP*

AuxP is used to mark the postpositions (heads) of the postpositional phrases. The postposition will recieve the *AuxP* label and the element attached to *AuxP* will receive the afun *(AAdjn , AComp, Atr, Comp)* according to their context of occurrence. The Figures 3.13 & 3.14 show examples for *AuxP*.



Tamil:	தனித்துப்	போட்டியிடுவது	தொடக்கத்தில்	சிரமமானது	என்று	எனக்குத்	தெரியும்	.
Tr:	TaniTTup	pOttiyituvaTu	TotakkaTTil	ciramamAnaTu	**enRu**	enakkuT	Teriyum	.
Gloss:	alone	competing	in-the-beginning	difficult	**that**	I	know	.
English:	I know **that** it is difficult to compete alone in the beginning .							

Figure 3.9: *AuxC:* Subordinating conjunctions



Tamil:	கொழும்பு	திரும்பியவ்	உடன்	புதன்கிழமை	மாலை	தமிழ்	எம்பிக்க்ள்	மற்றும்	தமிழர்	கட்சித்	தலைவர்கள்	உடன்	அவர்	ஆலோசனை	செய்வார்	என்று	தெரிகிறது	.
Tr:	kozumpu	Tirumpiyav	**utan**	puTankizamai	mAlai	Tamiz	empikkaL	maRRum	Tamizar	katciT	TalaivarkaL	utan	avar	AlOcanai	ceyvAr	**enRu**	TerikiRaTu	.
Gloss:	Colombo	return	**as-soon-as**	Wednesday	evening	Tamil	MPs	and	Tamil	party	leaders	with	he	discussion	do-he	**that**	seems	.
English:	It seems **that** she will discuss with Tamil MPs and Tamil party leaders **as soon as** she returns to Colombo .																	

Figure 3.10: *AuxC:* Subordinating conjunctions

3.3.11 Afun: *AuxS*

AuxS is used to label the technical root of a tree. In the Figure 3.14, the technical root (shown as *StaA*) is the root of the tree structure. To this node, the predicate and the sentencing ending node will be attached.

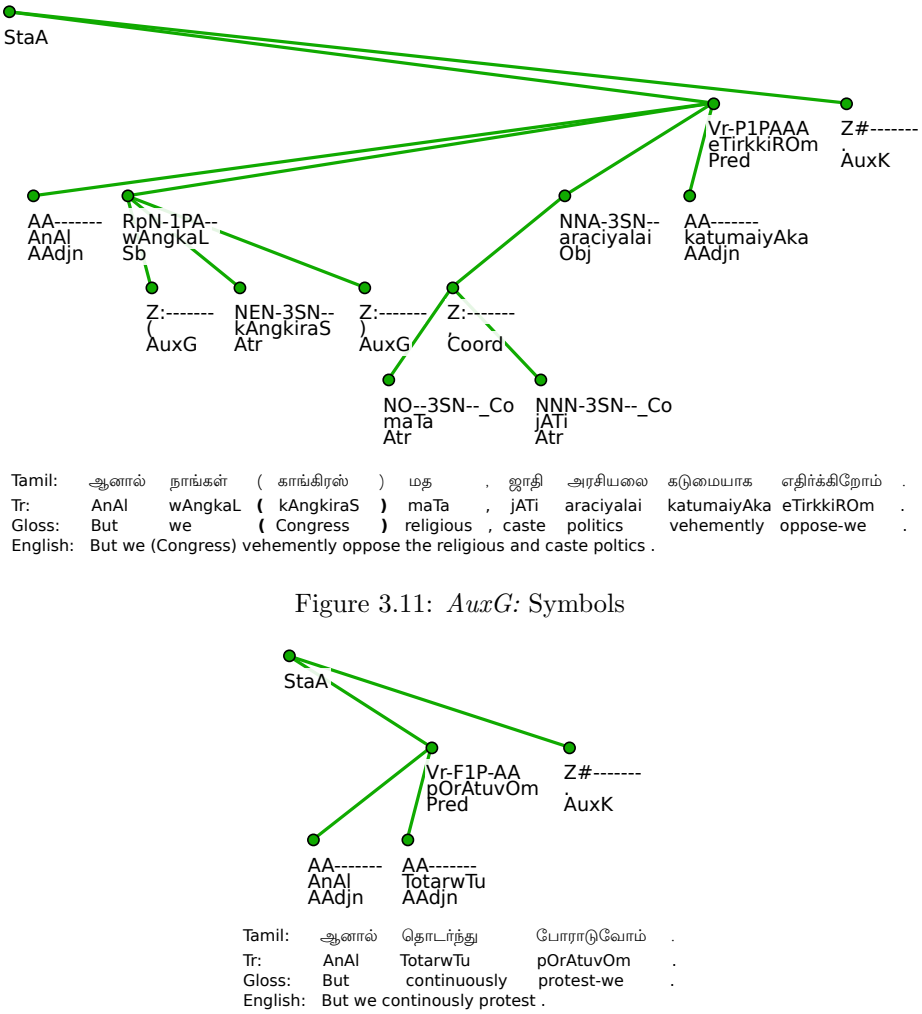


Figure 3.11: *AuxG:* Symbols

Figure 3.12: *AuxK:* Sentence termination symbol

3.3.12 Afun: *AuxV*

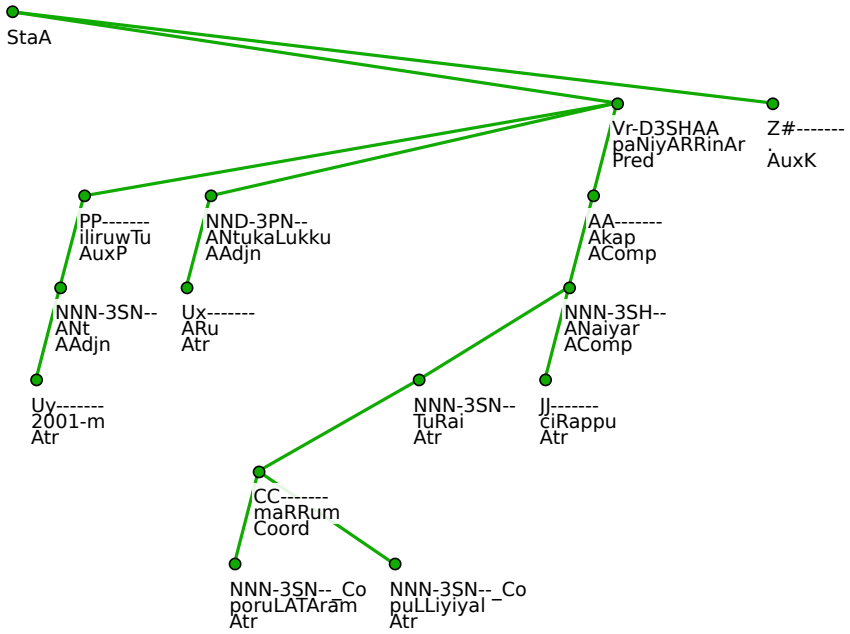
Auxiliary verbs are assigned the afun *AuxV*. In compound verb constructions, the auxiliary verb will be hanged under lexical verb. All auxiliary words including passive constructions will receive tha afun *AuxV*. In the Figure 3.15, there are two auxiliary verbs *patu* ('experience') and *uL* ('exist'). The auxiliary *patu* ('experience') and *uLLaTu* ('exist') are labeled with *AuxV*. The lexical verb receive the label *Pred* if there is only one clause in the sentence, otherwise the label of the lexical verb depends on the upper clauses.

3.3.13 Afun: *AuxX*

AuxX afun is used to label commas. All commas except the commas which act as coordination head is labeled with *AuxX*. The Figure 3.16 shows how *AuxX* has been annotated in the coordination structure.

3.3.14 Afun: *AuxZ*

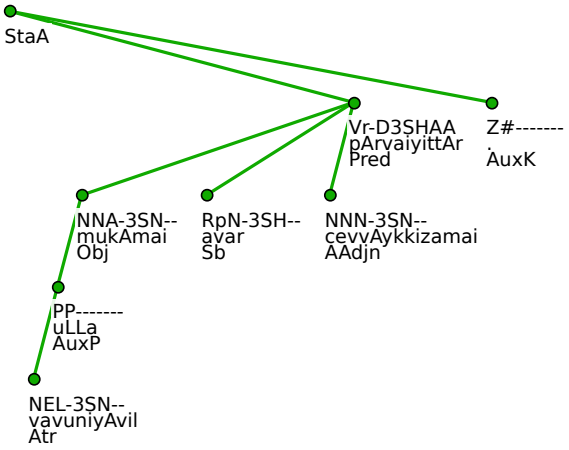
The clitics *um, E, TAn, mattum* will receive the afun *AuxZ*. The Figures 3.17 & 3.18 illustrate the labeling of *AuxZ*. The afun *AuxZ* corresponds to emphasizing words in English such as ('just', 'only', 'indeed') etc. These are not separate words but clitics in Tamil. Among the clitics, *um* has various semantics functions including acting as *coordination head*. In the Figure 3.17, the clitic *um* ('also') adds an inclusive meaning to the sentence. In the Figure 3.18, the clitic *TAn* ('only') emphasizes the entire clause.



Tamil:	2001-ம்	ஆண்ட்	இலிருந்து	ஆறு	ஆண்டுகளுக்கு	பொருளாதாரம்	மற்றும்	புள்ளியியல்	துறை	சிறப்பு	ஆணையர்	ஆகப்	பணியாற்றினார்	.
Tr:	2001-m	ANt	**iliruwTu**	ARu	ANtukaLukku	poruLATAram	maRRum	puLLiyiyaL	TuRai	ciRappu	ANaiyar	Akap	paNiyARRinAr	.
Gloss:	2001	year	**from**	six	years-for	economics	and	statistics	department	special	commissioner	as	worked-he	.

English: He/she worked as special commissioner of economics and statistics department for six years from 2001 .

Figure 3.13: *AuxP:* Postpositions



Tamil:	வவுனியாவில்	உள்ள	முகாமை	அவர்	செவ்வாய்க்கிழமை	பார்வையிட்டார்	.
Tr:	vavuniyAvil	**uLLa**	mukAmai	avar	cevvAykkizamai	pArvaiyittAr	.
Gloss:	in-Vavunia	**exist**	camp	he/she	Tuesday	visited	.

English: He/she visited the camp **in** Vavunia on Tuesday.

Figure 3.14: *AuxP:* Postpositions

3.3.15 Afun: *CC*

The label *CC* is used to mark in places where a single lexical unit is composed of multiple words. The *CC* relation of a word would indicate that the current word together with it's parent word form a single lexical unit. In Tamil, a single action verb can be split into multiple words. But during annotation, the problem arises as to which part of the word sequence the arguments of that lexical unit should be attached. To resolve this issue, we treat the first word as having the lexical meaning and the remaining to be children of the first word. In that case, the first word would receive the afun corresponding to the entire lexical unit and the remaining words would receive the label *CC*. For ex, the Figure 3.19 indicates that the Tamil verb *veRRipeRu* ('to win') is splitted into 2 words - *veRRi peRu*. The first word *veRRi* receives the afun which is meant for the entire lexical unit comprising the following words with afun *CC*.

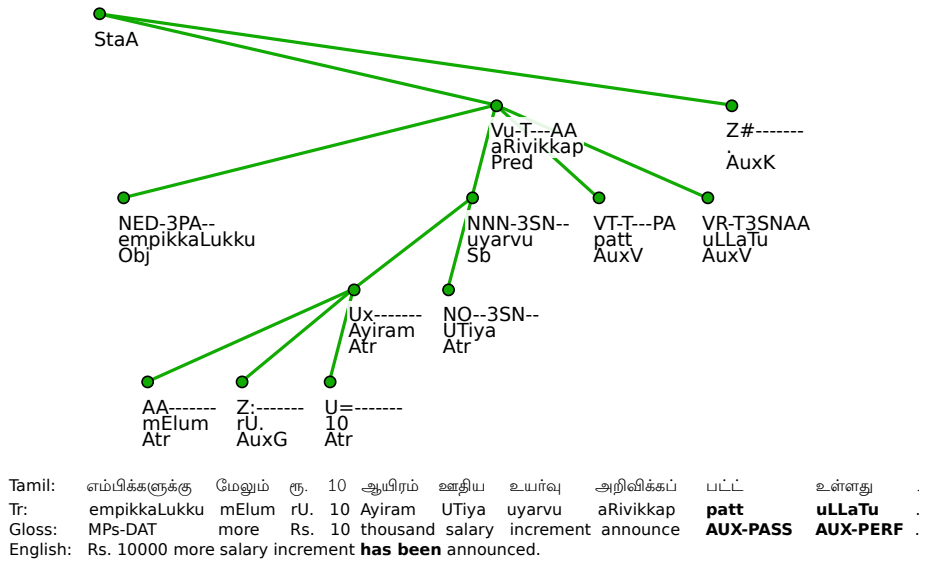


Figure 3.15: *AuxV:* Auxiliary verb

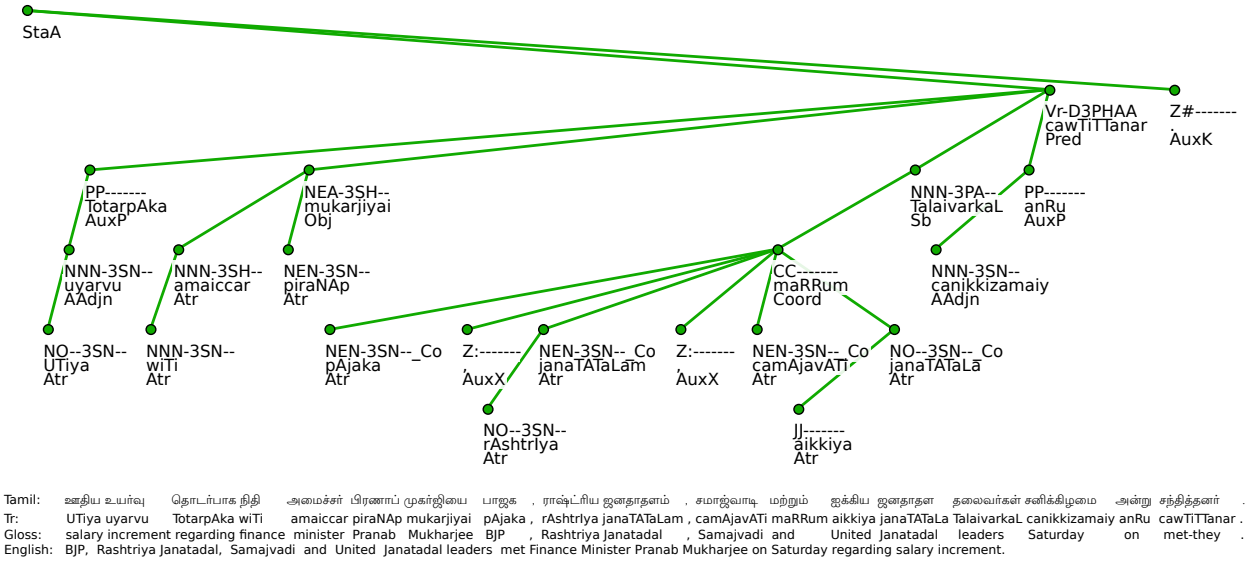
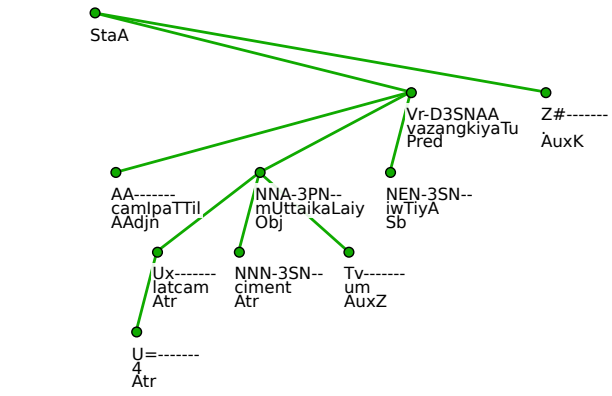


Figure 3.16: *AuxX:* Commas

Note:

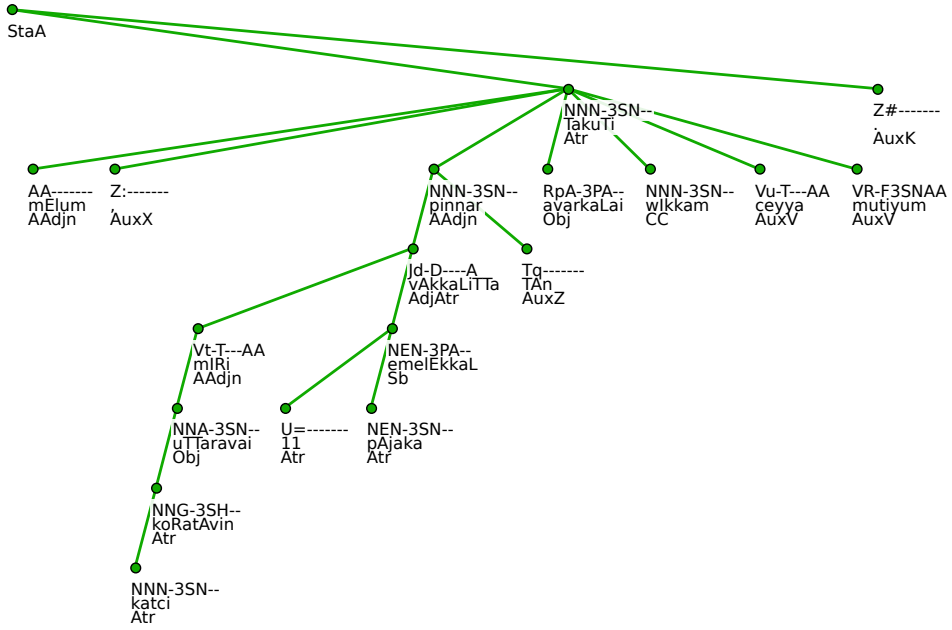
- The afun *CC* is not a relation for Multi Word Expressions (MWE).
- In many cases, Tamil verbs are formed using *noun + auxiliary* combination which acts as a single verb. In this case, the auxiliary would receive *CC* and would be attached under the noun.
- The word sequence can be written together as a single word. This implies that the word sequence is not a MWE.



Tamil:	சமீபத்தில்	4	லட்சம்	சிமெண்ட்	மூட்டைகளைய்	உம்	இந்தியா	வழங்கியது	.
Tr:	camIpaTTil	4	latcam	cimeNt	mUttaikaLy	**um**	iwTiyA	vazangkiyaTu	.
Gloss:	recently	4	lakhs	cement	packets	**also**	India	provided	.

English: Recently, India **also** provided 4 lakhs cement packets.

Figure 3.17: *AuxZ:* Emphasis



Tamil:	...	பாஜக	எமெலேக்கள்	வாக்களித்த	பின்னர்	தான்	அவர்களை	தகுதி	நீக்கம்	செய்ய	முடியும்	.
Tr:	...	pAjaka	emelEkkaL	vAkkaLiTTa	pinnar	**TAn**	avarkaLai	TakuTi	wIkkam	ceyya	mutiyum	.
Gloss:	...	BJP	MLAs	vote	after	**only**	them-ACC	disqualify	-	can	AUX	.

English: .... **Only** after BJP MLAs vote, it is possible to disqualify them .

Figure 3.18: *AuxZ:* Emphasis

3.3.16 Afun: *Comp*

Comp label is used to mark the obligatory element not attaching to verbs. For example, consider the phrase *1200kk um mERpatta poTumakkaL uyirizawT uLLanar* ("more than 1200 people have been died"), in that phrase, *1200kk* occurs as an obligatory argument to *mERpatta* ('more than'). So, *1200kk* will be labeled with *Comp*. Even nouns (not modifiers) which obligatorily attach to other nouns are labeled with *Comp* afun. Other occurrences of *Comp* is when postpositional phrase (PP phrase) attaches to a noun phrase. The postpositional head will receive *AuxP* label whereas the head noun phrase of the PP phrase will receive *Comp* label. The Figure 3.20 illustrates the labeling *Comp* relation.

3.3.17 Afun: *Coord*

Coordination is one of the complex phenomena in Tamil. Coordination conjunction in Tamil can be performed using at least 2 different ways. In the first method (for 'and' coordination), all conjoining

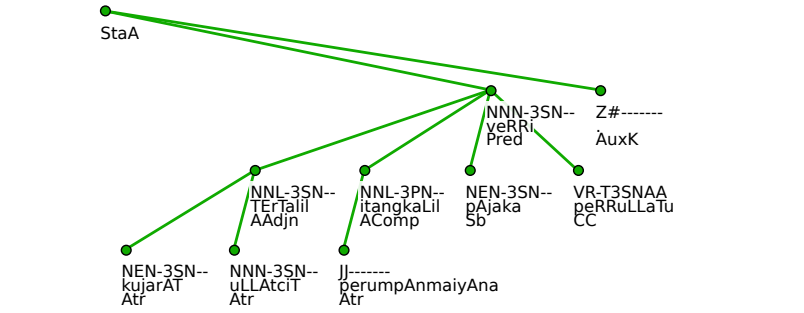


Figure 3.19: *CC:* Marker for a multi word sequence

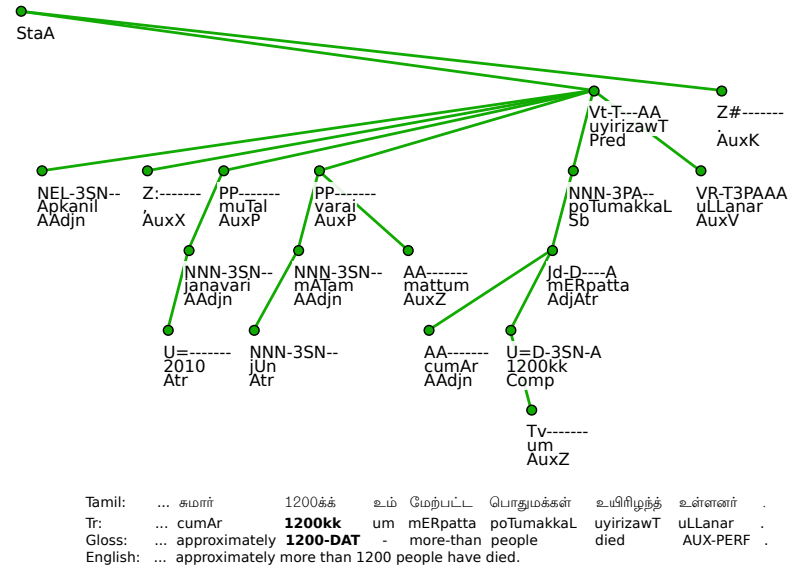


Figure 3.20: *Comp:* Complement (other than verbs)

elements adds the inclusive particle *um* ('also') at the end of the word form. Thus in this method, all conjoining elements possess the suffix (um) which would indicate the coordination is taking place. Moreover, the `'is_member'` attribute of the conjoining elements will be set to 1. The separator (comma) between elements is optional. It is perfectly legitimate if there is no comma between any of the conjoining elements.

Second method for Tamil 'and' coordination is similar to English style 'and' coordination. The conjunction word *maRRum* ('and') is added between conjoining elements. If there are more than 2 elements, then *maRRum* ('and') will be added just before the last conjunct. The other elements will be separated by comma. Again, the `'is_member'` attribute of the conjoining elements will be set to 1.

The 'or' coordination is performed in a similar way for both the methods. For the first method, the suffix *-O* is added to all conjuncts, and for the second method, the conjunction word *allaTu* ('or') is added between the last 2 conjoining elements. The remaining elements will be separated using comma.

Apart from the above two main methods, the coordination can be done via with just commas. In that case, the comma between conjuncts will act as coordination head. The Figures 3.21, 3.22 and 3.23 illustrates how coordination head is marked in the above mentioned scenarios.

3.3.18 Afun: *Obj*

Direct and indirect objects receive the afun *Obj*. The Figure 3.24 shows an example annotation of sentence with *Obj* relation. If there are both direct and indirect objects in a same sentence, then both will be labeled with afun *Obj*.

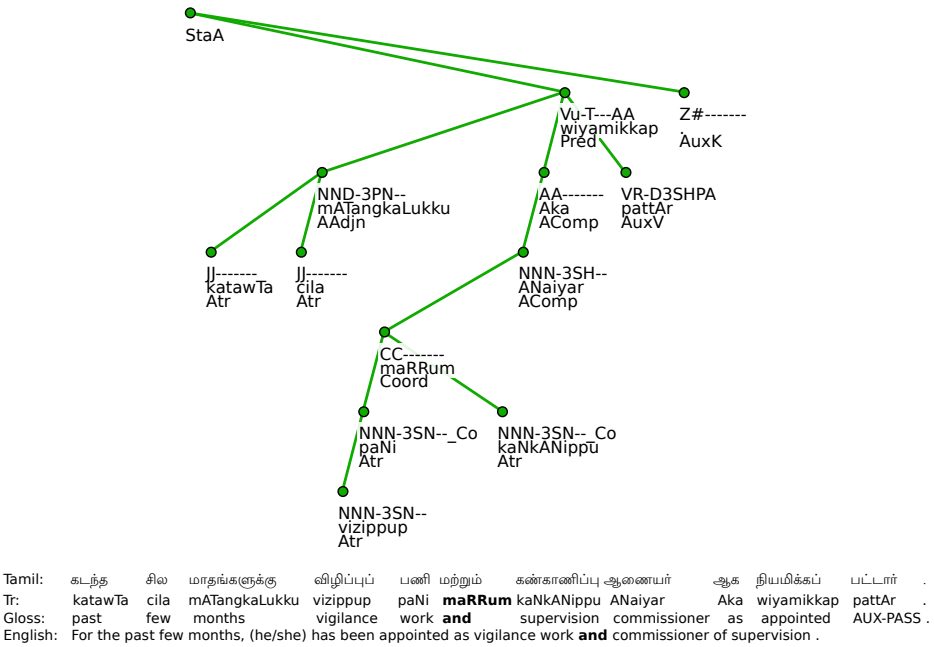


Figure 3.21: *Coord:* Coordination head (English style)

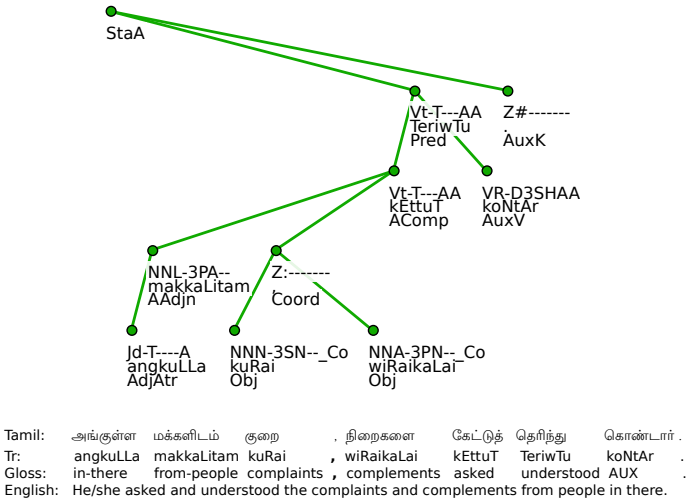


Figure 3.22: *Coord:* Coordination head (Comma)

3.3.19 Afun: *Pnom*

Nominal predicate occurs in the copula (be) constructions or verbless constructions. In these constructions, the sentence will not have any lexical verb. Instead, the predicate will contain only noun phrase. The noun phrase will be the predicate, and the afun label *Pnom* will be assigned to the noun phrase. The Figure 3.25 shows the usage of *Pnom*.

3.3.20 Afun: *Pred*

Predicate of the main clause will be given Pred. Only fnite verbs in Tamil can be the predicate of the sentence. In Tamil, fnite verbs at the end of the sentences are main predicates. So they receive Pred afun. In some cases, finite verbs will be absent at the end of the sentence. In that case, if there is a nominal predicate, it will receive Pnom afun or there won't be any Pred for that sentence. The Figure 3.26 shows an example annotation for *Pred* relation.

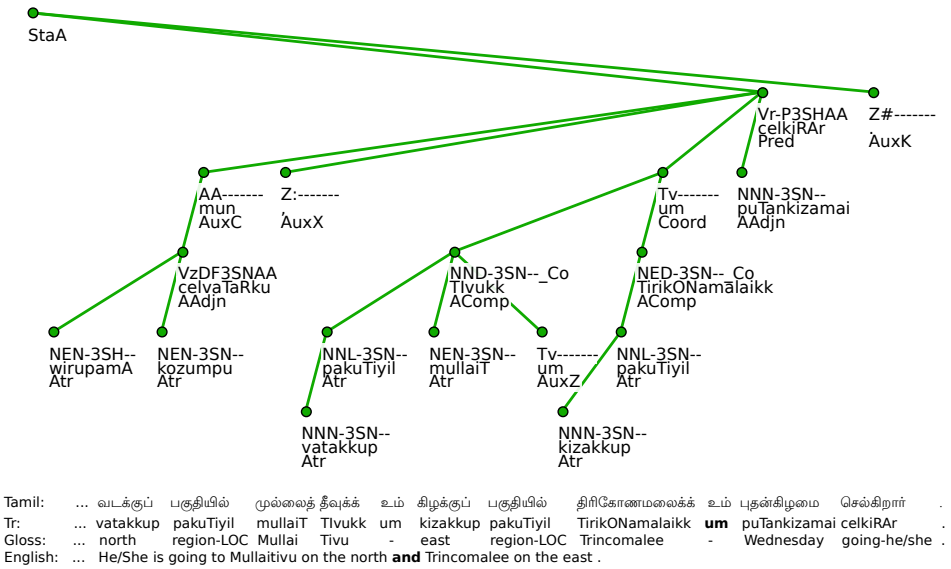


Figure 3.23: *Coord:* Coordination head (Morphological marker *-um*)

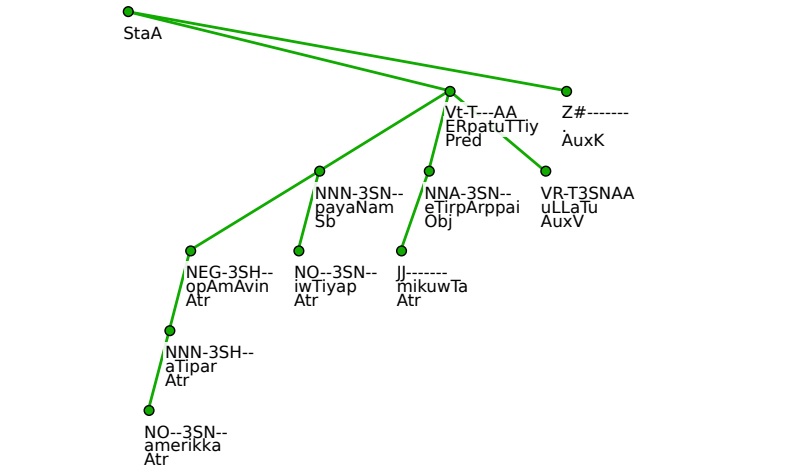


Figure 3.24: *Obj:* Object

3.3.21 Afun: *Sb*

The label *Sb* is assigned to the subject of the sentence. If there are more than one subjects in the sentence (i.e in the case of multiple clauses), then the label *Sb* will be assigned to all of them. The subject of passive verbs will also be labeled with *Sb* relation. The Figure 4.21 shows the example usage of Sb.

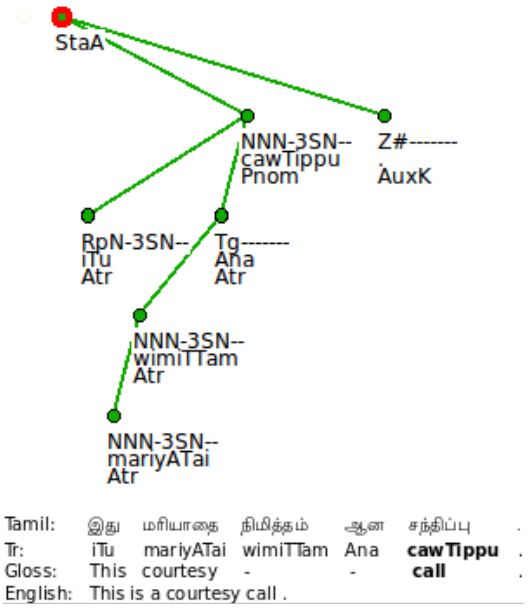


Figure 3.25: *Pnom:* Pnom

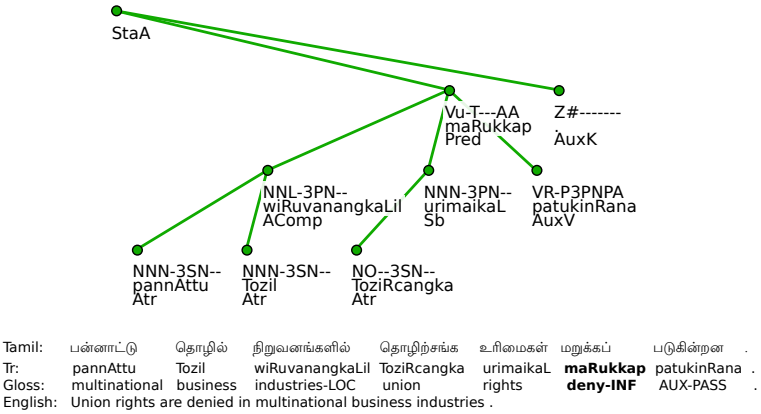


Figure 3.26: *Pnom:* Pred

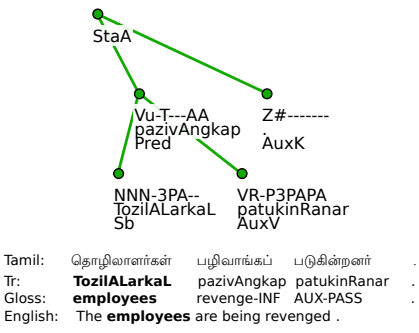


Figure 3.27: *Sb:* Subject

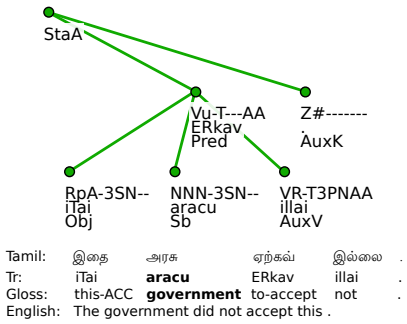


Figure 3.28: *Sb:* Subject

Acknowledgements

This project has been supported by,

- The European Commission's 7th Framework Program (FP7) under grant agreement n° 238405 (CLARA)[1] and

- The Grant MSM 0021620838 of the Czech Ministry of Education.

[1] CLARA Homepage: http://clara.uib.no/

Bibliography

[Begum et al., 2008] Begum, R., Husain, S., Dhwaj, A., Sharma, D. M., Bai, L., and Sangal, R. (2008). Dependency Annotation Scheme for Indian Languages. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, IJCNLP 2008, pages 721–726. Asian Federation of Natural Language Processing (AFNLP).

[Bharati et al., 2009] Bharati, A., Gupta, M., Yadav, V., Gali, K., and Sharma, D. M. (2009). Simple parser for Indian languages in a dependency framework. In *Proceedings of the Third Linguistic Annotation Workshop*, ACL-IJCNLP '09, pages 162–165, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Dhanalakshmi et al., 2010] Dhanalakshmi, V., Anand Kumar, M., Rekha, R., Soman, K., and Rajendran, S. (2010). Grammar Teaching Tools for Tamil Language. In *Technology for Education Conference (T4E 2010)*.

[Janarthanam et al., 2007] Janarthanam, S., Nallasamy, U., Ramasamy, L., and Santhoshkumar, C. (2007). Robust Dependency Parser for Natural Language Dialog Systems in Tamil. In *Proceedings of the 5th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, IJCAI KRPDS-2007, pages 1–6.

[Lehmann, 1989] Lehmann, T. (1989). *A Grammar of Modern Tamil*. Pondicherry Institute of Linguistics and Culture.

[Nivre, 2009] Nivre, J. (2009). Parsing Indian Languages with MaltParser. In *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, ICON 2009, pages 12–18.

[Ramasamy and Žabokrtský, 2011] Ramasamy, L. and Žabokrtský, Z. (2011). Tamil Dependency Parsing: Results Using Rule Based and Corpus Based Approaches. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part I*, CICLing'11, pages 82–95, Berlin, Heidelberg. Springer-Verlag.

[Selvam et al., 2009] Selvam, M., Natarajan, A. M., and Thangarajan, R. (2009). Structural Parsing of Natural Language Text in Tamil Language Using Dependency Model. *Int. J. Comput. Proc. Oriental Lang.*, 22(2-3):237–256.

[UFAL, 2006] UFAL (2006). The Prague Dependency Treebank 2.0.

[Vempaty et al., 2010] Vempaty, C., Naidu, V., Husain, S., Kiran, R., Bai, L., Sharma, D. M., and Sangal, R. (2010). Issues in Analyzing Telugu Sentences towards Building a Telugu Treebank. In Gelbukh, A. F., editor, *CICLing*, volume 6008 of *Lecture Notes in Computer Science*, pages 50–59. Springer.

[Žabokrtský et al., 2008] Žabokrtský, Z., Ptáček, J., and Pajas, P. (2008). TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 167–170, Stroudsburg, PA, USA. Association for Computational Linguistics.

The ÚFAL/CKL Technical Report Series

ÚFAL

ÚFAL (Ústav formální a aplikované lingvistiky; http://ufal.mff.cuni.cz) is the Institute of Formal and Applied linguistics, at the Faculty of Mathematics and Physics of Charles University, Prague, Czech Republic. The Institute was established in 1990 after the political changes as a continuation of the research work and teaching carried out by the former Laboratory of Algebraic Linguistics since the early 60s at the Faculty of Philosophy and later the Faculty of Mathematics and Physics. Together with the "sister" Institute of Theoretical and Computational Linguistics (Faculty of Arts) we aim at the development of teaching programs and research in the domain of theoretical and computational linguistics at the respective Faculties, collaborating closely with other departments such as the Institute of the Czech National Corpus at the Faculty of Philosophy and the Department of Computer Science at the Faculty of Mathematics and Physics.

CKL

As of 1 June 2000 the Center for Computational Linguistics (Centrum komputační lingvistiky; http://ckl.mff.cuni.cz) was established as one of the centers of excellence within the governmental program for support of research in the Czech Republic. The center is attached to the Faculty of Mathematics and Physics of Charles University in Prague.

Technical Reports

The ÚFAL/CKL technical report series has been established with the aim of disseminate topical results of research currently pursued by members, cooperators, or visitors of the Institute. The technical reports published in this Series are results of the research carried out in the research projects supported by the Grant Agency of the Czech Republic, GAČR 405/96/K214 ("Komplexní program"), GAČR 405/96/0198 (Treebank project), grant of the Ministry of Education of the Czech Republic VS 96151, and project of the Ministry of Education of the Czech Republic LN00A063 (Center for Computational Linguistics). Since November 1996, the following reports have been published.

ÚFAL TR-1996-01 Eva Hajičová, *The Past and Present of Computational Linguistics at Charles University*
Jan Hajič and Barbora Hladká, *Probabilistic and Rule-Based Tagging of an Inflective Language – A Comparison*

ÚFAL TR-1997-02 Vladislav Kuboň, Tomáš Holan and Martin Plátek, *A Grammar-Checker for Czech*

ÚFAL TR-1997-03 Alla Bémová at al., Anotace na analytické rovině, *Návod pro anotátory (in Czech)*

ÚFAL TR-1997-04 Jan Hajič and Barbora Hladká, *Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structural Tagset*

ÚFAL TR-1998-05 Geert-Jan M. Kruijff, *Basic Dependency-Based Logical Grammar*

ÚFAL TR-1999-06 Vladislav Kuboň, *A Robust Parser for Czech*

ÚFAL TR-1999-07 Eva Hajičová, Jarmila Panevová and Petr Sgall, *Manuál pro tektogramatické značkování (in Czech)*

ÚFAL TR-2000-08 Tomáš Holan, Vladislav Kuboň, Karel Oliva, Martin Plátek, *On Complexity of Word Order*

ÚFAL/CKL TR-2000-09 Eva Hajičová, Jarmila Panevová and Petr Sgall, *A Manual for Tectogrammatical Tagging of the Prague Dependency Treebank*

ÚFAL/CKL TR-2001-10 Zdeněk Žabokrtský, *Automatic Functor Assignment in the Prague Dependency Treebank*

ÚFAL/CKL TR-2001-11 Markéta Straňáková, *Homonymie předložkových skupin v češtině a možnost jejich automatického zpracování*

ÚFAL/CKL TR-2001-12 Eva Hajičová, Jarmila Panevová and Petr Sgall, *Manuál pro tektogramatické značkování (III. verze)*

ÚFAL/CKL TR-2002-13 Pavel Pecina and Martin Holub, *Sémanticky signifikantní kolokace*

ÚFAL/CKL TR-2002-14 Jiří Hana, Hana Hanová, *Manual for Morphological Annotation*

ÚFAL/CKL TR-2002-15 Markéta Lopatková, Zdeněk Žabokrtský, Karolína Skwarská and Vendula Benešová, *Tektogramaticky anotovaný valenční slovník českých sloves*

ÚFAL/CKL TR-2002-16 Radu Gramatovici and Martin Plátek, *D-trivial Dependency Grammars with Global Word-Order Restrictions*

ÚFAL/CKL TR-2003-17 Pavel Květoň, *Language for Grammatical Rules*

ÚFAL/CKL TR-2003-18 Markéta Lopatková, Zdeněk Žabokrtský, Karolina Skwarska, Václava Benešová, *Valency Lexicon of Czech Verbs VALLEX 1.0*

ÚFAL/CKL TR-2003-19 Lucie Kučová, Veronika Kolářová, Zdeněk Žabokrtský, Petr Pajas, Oliver Čulo, *Anotování koreference v Pražském závislostním korpusu*

ÚFAL/CKL TR-2003-20 Kateřina Veselá, Jiří Havelka, *Anotování aktuálního členění věty v Pražském závislostním korpusu*

ÚFAL/CKL TR-2004-21 Silvie Cinková, *Manuál pro tektogramatickou anotaci angličtiny*

ÚFAL/CKL TR-2004-22 Daniel Zeman, *Neprojektivity v Pražském závislostním korpusu (PDT)*

ÚFAL/CKL TR-2004-23 Jan Hajič a kol., *Anotace na analytické rovině, návod pro anotátory*

ÚFAL/CKL TR-2004-24 Jan Hajič, Zdeňka Urešová, Alevtina Bémová, Marie Kaplanová, *Anotace na tektogramatické rovině (úroveň 3)*

ÚFAL/CKL TR-2004-25 Jan Hajič, Zdeňka Urešová, Alevtina Bémová, Marie Kaplanová, *The Prague Dependency Treebank, Annotation on tectogrammatical level*

ÚFAL/CKL TR-2004-26 Martin Holub, Jiří Diviš, Jan Pávek, Pavel Pecina, Jiří Semecký, *Topics of Texts. Annotation, Automatic Searching and Indexing*

ÚFAL/CKL TR-2005-27 Jiří Hana, Daniel Zeman, *Manual for Morphological Annotation (Revision for PDT 2.0)*

ÚFAL/CKL TR-2005-28 Marie Mikulová a kol., *Pražský závislostní korpus (The Prague Dependency Treebank) Anotace na tektogramatické rovině (úroveň 3)*

ÚFAL/CKL TR-2005-29 Petr Pajas, Jan Štěpánek, *A Generic XML-Based Format for Structured Linguistic Annotation and Its application to the Prague Dependency Treebank 2.0*

ÚFAL/CKL TR-2006-30 Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolařová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský, *Annotation on the tectogrammatical level in the Prague Dependency Treebank (Annotation manual)*

ÚFAL/CKL TR-2006-31 Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolařová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Petr Sgall, Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský, *Anotace na tektogramatické rovině Pražského závislostního korpusu (Referenční příručka)*

ÚFAL/CKL TR-2006-32 Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolařová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Petr Sgall,Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský, *Annotation on the tectogrammatical level in the Prague Dependency Treebank (Reference book)*

ÚFAL/CKL TR-2006-33 Jan Hajič, Marie Mikulová, Martina Otradovcová, Petr Pajas, Petr Podveský, Zdeňka Urešová, *Pražský závislostní korpus mluvené češtiny. Rekonstrukce standardizovaného textu z mluvené řeči*

ÚFAL/CKL TR-2006-34 Markéta Lopatková, Zdeněk Žabokrtský, Václava Benešová (in cooperation with Karolína Skwarska, Klára Hrstková, Michaela Nová, Eduard Bejček, Miroslav Tichý) *Valency Lexicon of Czech Verbs. VALLEX 2.0*

ÚFAL/CKL TR-2006-35 Silvie Cinková, Jan Hajič, Marie Mikulová, Lucie Mladová, Anja Nedolužko, Petr Pajas, Jarmila Panevová, Jiří Semecký, Jana Šindlerová, Josef Toman, Zdeňka Urešová, Zdeněk Žabokrtský, *Annotation of English on the tectogrammatical level*

ÚFAL/CKL TR-2007-36 Magda Ševčíková, Zdeněk Žabokrtský, Oldřich Krůza, *Zpracování pojmenovaných entit v českých textech*

ÚFAL/CKL TR-2008-37 Silvie Cinková, Marie Mikulová, *Spontaneous speech reconstruction for the syntactic and semantic analysis of the NAP corpus*

ÚFAL/CKL TR-2008-38 Marie Mikulová, *Rekonstrukce standardizovaného textu z mluvené řeči v Pražském závislostním korpusu mluvené češtiny. Manuál pro anotátory*

ÚFAL/CKL TR-2008-39 Zdeněk Žabokrtský, Ondřej Bojar, *TectoMT, Developer's Guide*

ÚFAL/CKL TR-2008-40 Lucie Mladová, *Diskurzní vztahy v češtině a jejich zachycení v Pražském závislostním korpusu 2.0*

ÚFAL/CKL TR-2009-41 Marie Mikulová, *Pokyny k překladu určené překladatelům, revizorům a korektorům textů z Wall Street Journal pro projekt PCEDT*

ÚFAL/CKL TR-2011-42 Loganathan Ramasamy, Zdeněk Žabokrtský, *Tamil Dependency Treebank (TamilTB) - 0.1 Annotation Manual*