Domain Adaptation of Tamil Syntactic Parser:

A Data-driven Approach

A thesis submitted to the University of Hyderabad
for the award of the degree of

Doctor of Philosophy
In
Applied Linguistics

by
Keerthana B.
Reg. No:16HAPH03

Supervisor
Dr. K. Parameswari

Center for Applied Linguistics and Translation Studies
School of Humanities, University of Hyderabad, Hyderabad, Telangana, India.
March, 2024

Center for Applied Linguistics and Translation Studies
School of Humanities
University of Hyderabad

CERTIFICATE

Dated - 13/03/2024

This is to certify that **Keerthana B.** has carried out the research-work embodied in the present thesis titled **"Domain Adaptation of Tamil Syntactic Parser: A Data-driven Approach"** at the Center for Applied Linguistics and Translation Studies, University of Hyderabad. This thesis represents her independent work and has not been submitted for any research degree of this university or any other university.

This thesis is free from plagiarism and has not been submitted in part or in full to this University or any other university or institution for the award of any degree or diploma.

The following papers were published during this period:

1. Keerthana B and Parameswari K. 2019. Towards building a dependency parser for Tamil: A discussion on tags. Tamil Internet Conference 2019 during 20-22 September 2019 at the Anna University Guindy-Chennai Campus.
2. Keerthana B. and Parameswari K. 2018. "Parsing in Indian Languages: With Special Reference to Tamil The State-of-the-Art". In Journal Dravidian Studies, Dravidian University: Kuppam. ISSN 0976-5182

Further, the student has passed the following courses towards the fulfillment of the coursework requirement of PhD.:

Course	Course name	Credits	Pass/Fail
AL-801	Research Methodology	4.00	Pass
EG-825	Academic Writing for Doctoral Students	4.00	Pass
AL-802	Current Trends in Applied Linguistics	4.00	Pass
AL-821	Readings in Applied Linguistics	4.00	Pass

Dr. K. Parameswari
Supervisor
Center for Applied Linguistics and Translation Studies
School of Humanities

Head of the Department
CALTS

Dean
School of Humanities
University of Hyderabad

Center for Applied Linguistics and Translation Studies
School of Humanities
University of Hyderabad

DECLARATION

I hereby declare that the work embodied in this thesis entitled "**Domain Adaptation of Tamil Syntactic Parser: A Data-driven Approach**" is carried out by me under the supervision of Dr. K. Parameswari, Centre for Applied Linguistics and Translation Studies, University of Hyderabad, Hyderabad, and has not been submitted for any degree in part or in full to this university or any other university. I hereby agree that my thesis can be deposited in Shodhganga/INFILBNET.

A report of plagiarism statistics from the Indira Gandhi Memorial Library, University of Hyderabad is enclosed.

Keerthana B.
16HAPH03
Dated: 13.03.2024

Dr. K. Parameswari
Supervisor
Centre for Applied Linguistics and Translation Studies
School of Humanities

Acknowledgement

The journey was very comfortable initially and I was very happy to work in the field of research. However, the journey became more serious and tiring. The journey of PhD was filled with ups and downs. This thesis is dedicated to my *ammā* who is no more with us and to my little daughter who is my lifeline.

I would like to express my deepest gratitude and appreciation to all those who have supported and guided me throughout the journey of completing this Ph.D. thesis.

First and foremost my supervisor, Dr. K. Parameswari, for her unwavering support, invaluable insights, and scholarly guidance. Her mentorship has been instrumental in shaping the direction of my research and academic growth. She was the only one who constantly motivated me to complete my thesis during my crisis time.

I extend my sincere thanks to the members of my doctoral committee, Prof.S.Arulmozi and Prof.Uma Maheshwar Rao for their feedback and suggestions that significantly contributed to the refinement of my work. I also extend my thanks to Murthy garu, the office staff for his responsible work in providing me all the documents on time.

My heartfelt thanks to the department of CALTS and The University of Hyderabad for providing a conducive research environment and a platform for intellectual growth. I am grateful for the mutual trust and collaborative spirit among my fellow graduate students.

A special note of thanks to my family and friends for their unwavering encouragement, understanding, and patience throughout the ups and downs of the doctoral journey. Their love and support have been my anchor, and I am truly fortunate to have them in my life. A very special mention to my *appā, citti* and husband who were my emotional support system.

A special mention to my colleagues, Albin Rico Xalxo, Prakash, Sangeetha, Krishna who did all the last minute work that I needed. A special mention to Mr. Nagaraj, Mrs. Prabha and Ms. Nisha for their unbelievable support in my annotation task. They were my constant pillars throughout my journey.

Last but not least, I express my gratitude to all those whose names may not appear here but who, in various ways, have contributed to my academic and personal growth.

Thank you all for being an integral part of this significant milestone in my life.

Transliteration Schema

Roman	a	ā	i	ī	u	ū	e	ē	ai	o	ō	au	H
Tamil	அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ	ஃ

Roman	ka	ṅa	ca	ña	ṭa	ṇa	ta	na	pa	ma	ya	ra	la	va	l̲a	ḷa	r̲a	n̲a
Tamil	க	ங	ச	ஞ	ட	ண	த	ந	ப	ம	ய	ர	ல	வ	ழ	ள	ற	ன

Roman	ja	sa	ṣa	ha
Tamil	ஜ	ஸ	ஷ	ஹ

Abbreviations

ACC	Accusative case
ADJ	Adjective
ADV	Adverb
ADP	Adposition
AI	Artificial Intelligence
ASS	Associative
AUX	Auxiliary
CAUS	Causative marker
CCG	Combinatory Categorial Grammar
CFG	Context-free Grammars
COMP	Complementizer
COND	Conditional
CCONJ	Coordinating Conjunction
CoNLL	Computational Natural Language Learning
CoNLL-X	Conference on Computational Natural Language Learning
CONT	Continuous
COP	Copula
CRF	Conditional Random Field
DAT	Dative Case
DET	Determiner
DG	Dependency Grammar
DP	Dependency Parsing
F	Feminine
FUT	Future Tense
GEN	Genitive Case
GPSG	Generalized Phrase Structure Grammar
HDTB	Hindi Treebank
HIT-SCIR	Harbin Institute of Technologies Research Center for Social Computing and Information Retrieval
HPSG	Head-driven Phrase Structure Grammar
ID	Immediate Dominance
IIIT-H	International Institute of Information Technology-Hyderabad
INF	Infinitive
INTJ	Interjection
ISBN	Incremental Sigmoid Belief Networks
LAS	Labelled Attachment Score
LL	Left to Right (Leftmost derivation type)
LP	Linear Precedence
LR	Left to Right (Right most derivation type)

LS	Label scores
LSP	Lexicalized and Statistical Parsing
M	Masculine
MALT	MaltParser
MST	Maximum Spanning Tree Parser
MWTT	Modern Written Tamil Treebank
N	Noun/ Neuter
NER	NAmed Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
NOM	Nominative case
NOUN	Nouns
NP	Noun Phrase
NUM	Number
PART	Participle/ Particle
PCFG	Probabilistic Context-Free Grammars
PL	Plural
POS	Parts of Speech
PP	Prepositional Phrase
PRES	Present
PRON	Pronouns
PROPN	Proper Noun
PSG	Phrase Structure Grammar
PST	Past
PUD	Parallel Universal Dependencies
PUNCT	Punctuation
RASP	Robust Accurate Statistical Parsing
S	Sentence
SCONJ	Subordinating Conjunction
SG	Singular
SOV	Subject Object Verb
SVM	Support Vector Machine
SYM	Symbols
TAG	Tree Adjoining Grammar
TAM	Tense Aspect Modal
TTB	Tamil Dependency Treebank
TTR	Type Token Ratio
UD	Universal Dependencies
UCREL	University Centre for Computer Corpus Research on Language
URL	Uniform Remote Locator

UTF	Unicode Transformation Format
V	Verb
VERB	Verbs
VP	Verb Phrase
WCDG	Weighted Constraint Dependency Grammar

Abstract

Parsing has been a highly spoken topic in recent years and attracted the interest of Natural Language Processing (NLP) researchers around the world. It is challenging when the language under study is a free-word order language and morphologically rich like Tamil. Parsing refers to the process of syntactic analysis of a specific language text. A parser is an automated tool that dissects sentences to provide syntactic/syntactico-semantic analysis of relations of words in a sentence. Parsing is useful in the downstream analysis and applications of NLP such as machine translation, document classification, dialogue modeling, etc.

This study adopts a data-driven approach for building a domain-specific parser for Tamil using domain-specific data. Adapting the existing IIIT-H Syntactic parser and extending the study to domain-specific data using UD framework is the current study. A detailed description of the tags used in morph, POS and syntactic relations are presented in this work. An enhanced annotation for language-specific features is included in this work. Challenges faced in parsing ambiguous domain-specific structures are elaborated.

The research further provides results, suggesting that enriching the current parser with more number of domains can increase the accuracy and tackle ambiguity better than existing one. Results are inspiring and this parser proves to be efficient for languages like Tamil which can be later extended to other morphologically-rich languages.

Table of Contents

Chapter 1

Introduction

1.1. What is parsing?

Parsing is a process of analyzing any language text syntactically. A parser is an automated programmed tool which segments a string of natural language data into words and gives the morphological, and syntactic tags in a sentence. Parsing plays a pivotal role in Artificial Intelligence (AI). Tools like Machine Translation systems, automated data extraction systems, Document Classification and Dialogue Modelling are some of its applications (Clark et al., 2013). Parser is the most wanted module in Natural Language Understanding (NLU) as it involves the understanding of the language's text. Parsing is an intricate task as it includes resolving ambiguities like attachment[1] and scope[2] ambiguities. The current work on Tamil parsing aims to build a treebank for Tamil using domain based corpus in Universal Dependencies framework (Nivre, 2009).

The Latin word "*pars*" meaning part/orationis is from where the word parsing originated. The word with the above meaning was used till 1500s in common English. In 1700s, the meaning was updated to 'analyze critically'. Later, in linguistics, Harry Bunt, John Carroll and Giorgio Satta (2005) defined parsing as "Parsing can be defined as the decomposition of complex structures into their constituent parts and parsing technology as the methods, the tools, and the software to parse automatically". In such traditional syntax, parsing refers to understanding the meaning of the sentence with the help of tree

[1] Attachment ambiguity refers to uncertainty in attachment and grammatical association of certain words in a sentence. Classic example is 'I saw a man with the telescope'. The possessor of the telescope is ambiguous.
[2] Scope ambiguity refers to uncertainty in the usage of quantifiers and other operators. Classic example is 'Every student didn't pass the exam'. The quantifier 'every' is ambiguous.

diagrams, highlighting the subject-predicate distinction. Whereas, in modern Computational linguistics, the term parsing is defined as "to analyze the input sentence in terms of grammatical constituents, identifying the parts of speech, syntactic relations". Parsing is a process of determining how a string of terminals (sentence) is generated from its constituents, by breaking down sentence into tokens (Rangra, R., 2015). In such cases, the string of words are analyzed to get a parsed tree as output, with syntactic and semantic meanings as well. Syntactic ambiguities are also seen in the output trees. This can be better understood by representing the data using dependency grammar as seen in figures 1.1 and 1.2.

(1.1) Radha wrote a letter

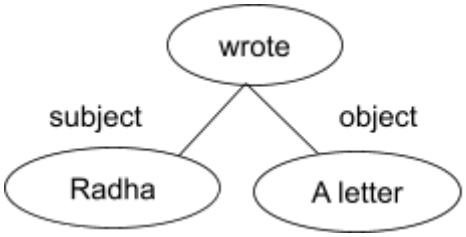


Figure 1.1: Structure representation of example 1.1

In the figure 1.1, the verb is the root of the sentence and Mary and a letter are dependents of the head (root). 'Mary' plays the role of subject and 'a letter' plays the role of object since the verb is transitive.

(1.2) *nāṉ* *vīṭṭuppāṭa.tt-ay.c* *cey-t-ēṉ*

I-NOM homework-ACC do-PST-1SG.M/F

'I did my homework'

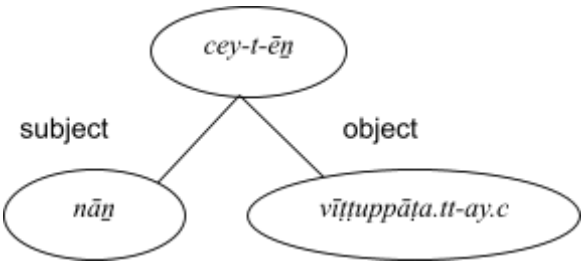


Figure 1.2: Structure representation of example 1.2

Example 1.2 takes the Tamil verb *ceytēṉ* as the root which has a subject *nāṉ* and an object *vīṭṭuppāṭam* respectively. The noticeable difference between the example 1.1 and 1.2 is the word order. English is a fixed order language unlike Tamil which is a free word-order language. Thus, the position of words is never related to the roles played by them. This becomes challenging and human resources play a pivotal role in resolving these ambiguities. Identification of good grammar and good parsing algorithm with respect to language specific demands is highly essential for a better output.

Developing a good Treebank[3] is an important benchmark in data-driven parsing. It is challenging when it comes to Dravidian languages including Tamil due to its agglutinative morphology and free word-order syntax. It is a burdensome task to include a huge variety of sentences that are prevalent in the language. This study takes up this challenge of building a treebank restricting to specific domains, as each domain has some unique syntactic constructions.

This study deals with domain adaptation of Tamil Syntactic Parser, IIIT Hyderabad[4]**.** This study adopts the guidelines and certain tools developed in the parser developed at IIIT-Hyderabad. A significant contribution was made on designing guidelines, on tagging POS and syntactic tags for the parser with regard to domain adaptation. It also studies various syntactic constructions present in different domains and presents the challenges posed by such data in parsing.

1.2. Aim and Objective of the Study

This research attempts to adapt an existing Tamil Syntactic parser for domain-specific data using the data-driven method. The following steps are investigated and worked upon to build a good domain-specific treebank

- To build an efficient data-driven Tamil syntactic parser which parses different domains of texts in Tamil

[3] Treebank- the term was coined in 1980s by Geoffrey Leech, which denotes the manually parsed tree structures on various levels of linguistic analysis
[4] http://10.2.4.118:3000/

- To choose the grammar formalism which would capture the language-specific features
- To study the existing mechanisms thoroughly and to choose the appropriate parsing technique
- To select domain based corpora for a better availability of a variety of sentential constructions.
- To annotate domain-specific treebanks and to identify various linguistic issues
- To compare the structural differences found across the domain
- To build domain adapted parser for Modern Tamil
- To evaluate and to find out efficient ways to improve the accuracy

1.3. An Overview of Tamil Syntax

Tamil is a south Dravidian language (Krishnamurti, 2003) spoken in Southern India and north-eastern SriLanka from prehistoric times (Ed.Steever,2020). In modern times, it is also spoken by a majority in countries like Malaysia, and Singapore and it is one of the official languages in Singapore and Sri Lanka. There are 69 million native speakers of Tamil in India (India, 2011). It is mainly spoken in the state of Tamil Nadu, followed by Puducherry, Kerala and Andaman and Nicobar islands. Tamil is a diglossic language and there are many dialects being spoken, some of which are very predominant. It includes Kongu dialect (covering Ooty, Coimbatore, Erode, Tiruppur, Salem, and Dindugal), Central Tamil dialect (spoken in Thanjavur, Tiruvarur, Nagapattinam, Karur and Tiruchirapalli), Madras Bashai (spoken in Chennai), Madurai Tamil, Nellai Tamil (Tirunelveli) and Kumari Tamil (Kanyakumari) in India. The thesis concentrates on the written Tamil to build the Tamil parser, as including all these dialects is a challenging task.

Tamil is a head-final, left-branching, Nominative-Accusative, pro-drop, genitive drop, copula drop language with Subject Object Verb (SOV) word order (Krsihnamurti, 2003). Verbless constructions are common occurrences in Tamil. A sentence can be either a verbal predicate or a nominal predicate. Nominal arguments or adjuncts and a verbal/nominal predicate are a part of Simple sentences in Tamil. Complex sentences are

a combination of a main clause and subordinate clauses (usually non-finite clauses). Compound or co-ordinate sentences are a combination of two main clauses with/without subordinate clauses. Subject argument is expressed using a nominative/ non-nominative case. Some structural features of Tamil are listed below:

- Tamil is an agglutinative language. In 1998, Annamalai and Steever stated that the "inflections are marked by suffixes attached to a lexical base, which may be augmented by derivational suffixes" (Ed.Steever, 2020). The complex linguistic information like the person, tense, aspect, number, gender information is encoded in suffixes.
  Example (1.3): *pār-ttu.k-koṇṭiru-nt-āḷ* 'see-PART-CONT-PST-3.SG.F.' '(she) was looking'

- Tamil is a pro-drop language that follows null-subject parameter. So, subjectless constructions are common in usage. "As finite verbs indicate the person, number and gender of the subject, the subject may be obviated without recourse to pronominal substitutes" (Ed.Steever, 2020).
  Example (1.4): *paṭi-tt-ēn* 'study-PST-1.SG.N' '(I) studied'

- Copulas are not mandatory (Ed. Steever, 2020).
  Example (1.5): *nāṉ āciriyar (āvēṉ)* 'I-NOM teacher (COP)' 'I am a teacher'

  Since copulas are optional, the constructions without copula are considered equative constructions, which are commonly found in Tamil. So, the predicate is realized as nominal or predicative adjectives.
  Example (1.6): *nāṉ uyaramāṉavaṉ* 'I-NOM tall' 'I am a tall person'

- Non-finite constructions, especially, infinitive constructions are very productive in Tamil. The infinitive marker in Tamil is *-a.*
  Example (1.7): *rām paṭi.kk-a.p pō-kiṟ-āṉ* 'Ram was about to study' (compound verb construction)/ 'Ram is going to study' (complex sentence construction)

Example (1.8) *[avaṉ var-a] nāṉ-um va.r- uv- ēṉ* ‘‘If he comes, I will also come’ (conditional clause)

Example (1.9) *nāṉ [paḷḷi- kku.p pōk-a] virumpu- kiṟ- ēṉ* ‘I like to go to school’ (complement to desiderative clause)

- Case syncretism, where a case marker has multiple functions. One such example in Tamil is that the dative case marker *-ku* has multiple functions. It can be used to denote indirect objects/ destinations/ goals/ dative case marked subjects/ to refer to a point or duration of time/ etc.

  Example (1.10): *rām appāvuk**ku** kaṭitam koṭuttāṉ* ‘Ram gave a letter to father’;

  (1.11): *nāṉ kōviluk**ku**c ceṉṟēṉ* ‘I went to temple’;

  (1.12): *eṉak**ku** avaṉ cāppiṭa vēṇṭum* ‘I want him to eat’.

- Non-nominative subjects like dative/ instrumental subjects are common occurrences found in Tamil.

  Example (1.13): *eṉak**ku** paṇam vēṇṭum* **‘I** want money’.

  Example (1.14): *uṉṉ**āl** vēlayyay ceyya muṭiyum* **‘You** can do the work’.

- NV compounds are very productive in Tamil. Nouns and verbs are found “as a sequence of words which occur together expressing a cohesive or unified meaning” (S. Rajendran, 2017).

  Example (1.15): *nīccal* (swimming-noun)-*aṭi* (beat-verb) ‘to swim’

- Participle constructions in complex sentences are common instances in Tamil. It is used to express simultaneous actions, consecutive actions, etc.

  Example (1.16): verbal participle: *muttu **vantu** ceṉṟāṉ* ‘Muthu came and went’

  Example (1.17): adjectival participle: ***vanta** māṇavaṉ* ‘The boy who came’

- Gerunds have multiple markers and are very productive in Tamil.

  Example (1.18): *-al*: *ceyt**al*** ‘doing’

  Example (1.19): *-atu*: *nīntuv**atu*** ‘swimming’

- Complements precede matrix clauses in all Dravidian languages including Tamil (Krshnamurti, 2003).
  Example (1.20): *avaṉ* ***varuvatāka*** *coṉṉāṉ* 'He said that he will come'

- Genitive case drop is possible without altering the meaning of the sentence.
  Example (1.21) *ravi* ***vīṭṭu/vīṭṭiṉ*** *arukil puttakattayp pārtēṉ* 'I saw the book near **Ravi's** house'

- Accusative cases can be marked optional on objects.
  Example (1.22) *malar oru* ***puttakam/ puttakattay*** *e[utiṉāl* 'Malar wrote a book'

1.4. Overview of Parsing

Parsing is an essential process involved in NLP that is used in AI engines. Even though it has been decades since NLP started, it remains inefficient for the machine to produce 100% efficiency and accuracy by itself. Human intervention is required to produce better results. Parsing requires a good annotation schema, a language-specific grammar formalism, right parsing strategy, and a proper implementation technique for achieving the best accuracy for any language. Each of these criteria are discussed in detail below.

1.4.1. A survey on grammar formalisms

Grammar formalisms are essential in building the annotation guidelines as they define the linguistic properties. They help in identifying the linguistic cues to automate the parsing tags. The most commonly used grammars are constituency and dependency approaches. However, other suitable grammar formalisms for building a parser are available including Generalized Phrase Structure Grammar, Head-driven Phrase Structure Grammar, Combinatory Categorial Grammar, Lexical Functional Grammar, Tree Adjoining Grammar, etc. Some of these are looked at in detail below.

1.4.1.1. Phrase Structure Grammar (PSG)

Phrase Structure Grammar, also called constituency grammar/ context-free grammar, was framed by Noam Chomsky (1959), where the words are clubbed together as constituents or phrases. To be precise, 'constituents' is defined as "a word or a group of words that functions as a single unit within a hierarchical structure" (Osborne, 2018). Following this definition, constituency grammar represents the syntactic structure of a sentence in terms of phrases. It is pictorially represented as trees, which clearly picturises the hierarchy of phrases in a sentence. Terminal nodes in such trees are the actual words of the sentence beyond which branching is not seen. Non-terminal nodes are branched nodes of the tree. It can be seen in tree representation 1.3 below:

1.3. Sita came to my home

[Sita]NP [went]VP[to my home]PP

This type of grammar is more suitable for fixed word-order languages rather than free word-order languages as the constituent order keeps varying in free word-order languages. English, a fixed word-order language, has done an extensive study on this grammar in NLP and treebanks like Penn Treebank works on this principle. However, it becomes challenging to group the constituents and frame a language specific-rule for free-word order languages.

1.4.1.2. Generalized Phrase Structure Grammar (GPSG)

GPSG is a constraint-based Phrase Structure Grammar, deriving from constituency grammar, developed by Gerald Gazdar in the 1970s with Ewan Klein, Ivan Sag, and Geoffrey Pullum for English (Cf. Gazdar, G., et.al, 1985). Unlike PSGs, GPSG has brought in multicomponent structures represented at the same hierarchy, which makes it more flexible and a real context-free grammar. This framework is formulated based on lexical rules. A computational perspective of GPSG was articled by Philips in 1992, stating the following rules: (i) Immediate-dominance (ID) states the possible combinations of grammatical categories to produce other categories; (ii) Linear Precedence (LP) states the linear order in which the constituents/phrases are arranged;

(iii) metarules are lexical redundancy rules which stick on to a specified pattern, also has some rules to tally the derived patterns (Ristad, E.S., 1989). GPSG is computationally implemented in languages including, English, Persian, French, Chinese and Arabic (Bahrani, M. et.al., 2011). An example of GSPG representation is given below:

1.4. Ram gave Sita a pencil

((NP-Ram (N)))((VP-gave (V)))((NP- Sita (N)))((NP-a (DET) pencil (N)))

1.4.1.3. Head-driven Phrase Structure Grammar (HPSG)

HPSG is a lexical- based, constrained PSG, developed by Carl Pollard and Ivan Sag (Cf. Pollard, C., and Sag, I. A., 1994), analyzes all the linguistic levels (phonology, morphology, syntax, semantics, and pragmatics) using feature-value pairings, structure sharing and relational constraints. This grammar's basic notion is obtained from Saussure's 'sign' (Müller, S.,et.al.,2021). In addition to research (such as grammar comparison and hypothesis testing), computational HPSG implementations find application in machine translation, question-answering, language tutoring, and other fields. Some of such implemented languages like Romance languages, Slavic languages, German, Japanese, Welsh, English, Korean and Warlpiri (Levine, R. D. and Meurers, W. D., 2006). *Enju* is one such famous parsing engine developed using HPSG in Japan. Hindi (Goyal, et al, 2003) and Bangla (Khan, Naira & Khan, Mumit., 2006) are the two Indian languages to implement HPSG. An illustration of HPSG is given below:

1.5. Felix chased the dog

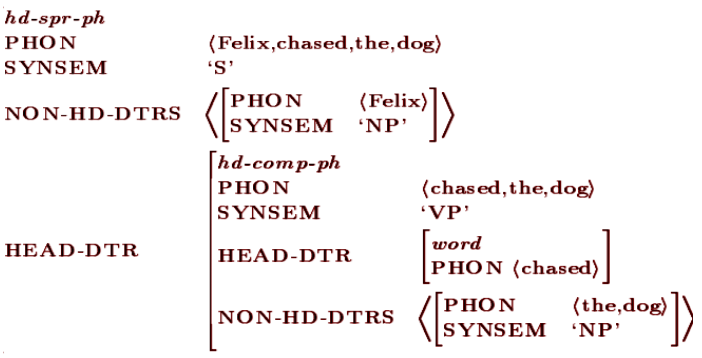


Figure 1.5. Sample HPSG tree representation is extracted from Sag, I. A., 1995:15

1.4.1.4. Combinatory Categorial Grammar (CCG)

A lexicalized grammar form where categorial grammar was extended with functional operators, was developed by Mark Steedman and Remo Pareschi (1987) and Szabolcsi (1992). This is one of the oldest grammars which consists of lexicon, pairing with lexical categories and a set of rules, suggesting the possible ways to combine the categories. The category is either atomic (S, N, NP, and PP) or complex (functors). Since there is no rule fixed for any category, the rules are determined only based on the provided input data, using precise mathematical computing of syntax. The rules are completely independent of any structure or pattern, which is very appropriate for free word-order languages.

It is applied in English parsing (Hockenmaier, J., and Steedman, M., 2002, 342); Korean parsing (Cha, J., Lee, G. & Lee, J., 2002); Chinese parsing (Tse, D. and Curran, J.R., 2010); English and Hindi parsing (Ambati, B.R., 2016 & Ambati, B.R., Deoskar, T. and Steedman, M., 2018) and Telugu tags are also improved using CCG tags (Kumari, B.V.S. and Rao, R.R., 2015).

An instance is illustrated below:

1.6. I dislike and Mary likes musicals

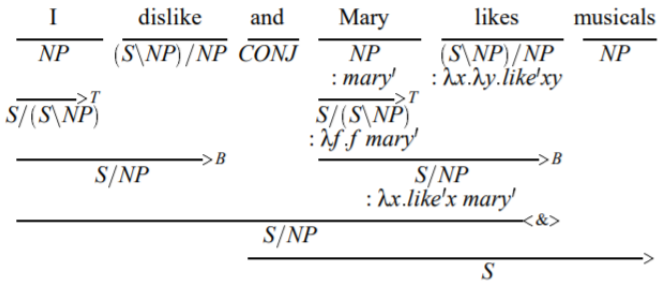


Figure 1.6: This sample representation is extracted from Mark Steedman, 1996 (4)

1.4.1.5. Lexical-Functional Grammar (LFG)

LFG was first published by Joan Bresnan (1982), which was represented in constituent and functional structure. It has distinct syntactic (c-structure) and semantic (f-structure) representations. LFG accommodates multiple interpretations of any syntactic word on f-structure, resulting in a context based tree structure. The main advantage of LFG is that the surface structure and the deep argument structure coincide with each other unlike Chomskyan's constituency parsing.

It is used in parsing Wall Street Journal (WSJ) by Stefan Riezler, et.al. (2002) with the f-score of 76.1%, Turkish by Güngördü, Z. and Oflazer, K., 1995, Sanskrit by Tapaswi, N., Jain, S. and Chourey, V., 2012. This was not very efficient as mapping between syntax and semantics was not very efficient.

1.7. He gave the woman the gift

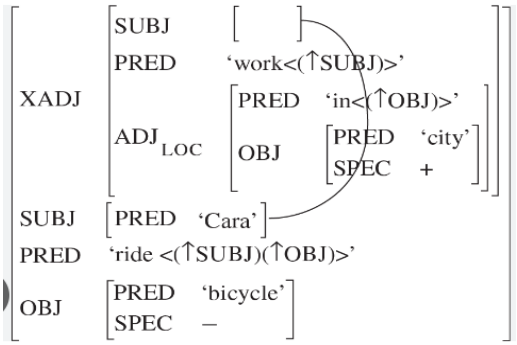


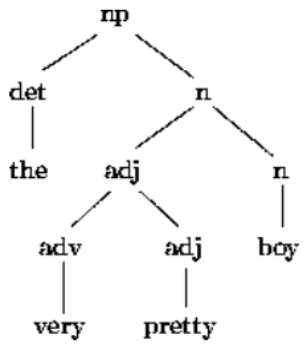
Figure 1.7. A reference example of LFG is extracted from Sells, P. (2013).

1.4.1.6. Tree Adjoining Grammar (TAG)

Tree Adjoining Grammar, formulated by Aravind Joshi (A. K. Joshi, Levy, and Takahashi, 1975) has both lexicalised and constraint-based variations. Basic parsed trees are combined with substitution and adjunction operations (Kroch, A. S., & Joshi, A. K., 1985) to describe natural language systems which have initial and auxiliary trees.

It is implemented in English and results obtained are better than previously mentioned formalisms (XTAG Research Group 1998; Abeille, A.; Bishop, K., Cote, Sharon, & Schabes, Y. 1990). Among Indian languages, Tamil TAG parsers were built by Menon, et al., in 2016; and Hindi TAG parser was built for text-scene conversion by Jain, et al. in 2018.

1.8. The very pretty boy



1.8. This sample representation is extracted from Tree Adjoining Grammars, (https://www.let.rug.nl/~vannoord/papers/diss/diss/node59.html)

1.4.1.7. Dependency Grammar (DG)

Dependency Grammar is one of the oldest theoretical and descriptive grammar, which can be traced back to Pāninian Sanskrit grammar. The modern thought of DG was proposed by a French linguist Lucien Tesnière (1959). Debusmann, R. (2000) reports that the term dependency's mathematical properties were first studied by Hays in 1964 and by Gaifman in 1965, whose aim was to develop an automated algorithm for parsing natural languages.

Unlike the other theories which focus on the constituents and their hierarchy, DG represents the relationship between the head and its dependents. Content words are tagged by the dependency relations and the functional words are related to the content words that they modify. Punctuations

attach to the head of the following or preceding phrase/ clause,depending on the punctuation used. This grammar is the pioneer of other variants such as, Functional Dependency Grammar, Operator Grammar, Word Grammar, Lexicase, Constraint Dependency Grammar, Extensible Dependency Grammar, Universal Dependencies, Link Grammar, etc,.

Dependency Grammar is well suitable for free-word order languages as the grammar does not segregate sentences into phrases. Also, verbs are considered the head of the sentence in DG. The number of dependency relations are equal to the number of words in a sentence, which makes DG a better computable grammar.

An comparative illustration on constituency structure and dependency structure is given below:

Example (1.19): Sam came to my home

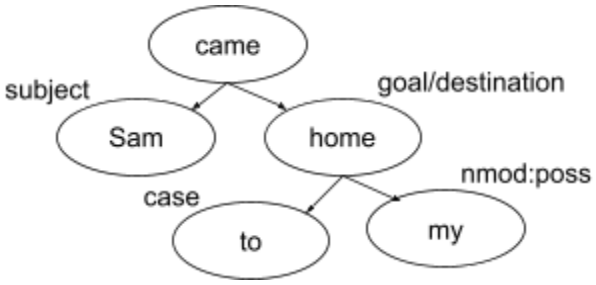


Fig 1.9: Dependency representation of example 1.19

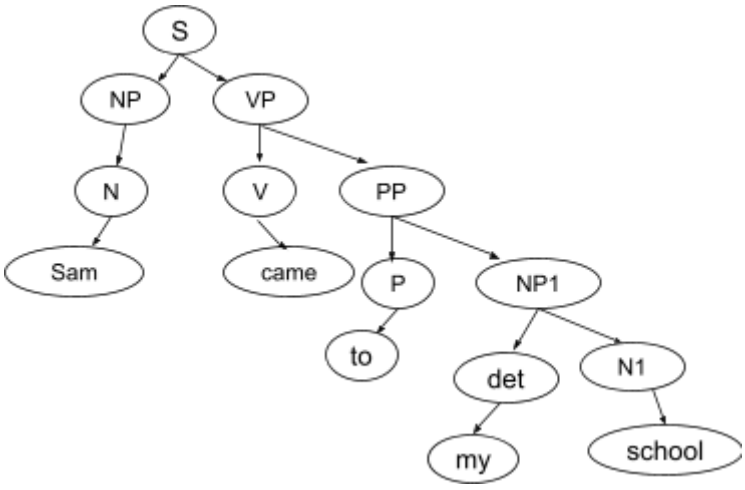


Fig 1.10: Constituency structure of example 1.19

From the above illustrations, it can be stated that DG is simpler to understand and easier to

annotate. Faster manual annotation and more efficient parsing is applicable for any language in DG (Jurafsky, D., & Martin, J. H., 2018). To be specific, for a morphologically rich and free word-order language like Tamil, implementing a dependency model is better (Falavarjani, S. A. M., & Ghassem-Sani, G., 2015).

A number of notable works have been implemented using DG in a number of Indian and other languages as discussed in chapter 2. The present study also uses the Dependency Grammar to build a Tamil parser.

1.4.2. Types of Parsing

Parsing is divided into two types, namely, top-down parsing and bottom-up parsing, based on the implementation of grammatical rules.

1.4.2.1. Top-down Parsing

Top-down parsing is a goal-oriented parsing technique which attempts to construct a parsed tree for the input from the root (top) to the leaves (bottom). The transitions of tokens are seen from left to right, attempting to resolve ambiguities by changing the rules of the right hand side. The major advantage of such systems is that it never wastes time in validating trees that it would not lead to S (root) but the negative aspect is that it processes output before examining the input (Cf. Jurafsky, 2000: 356-359). This type of parsing uses Context-free grammar, which is a set of rewriting rules.

Recursive descent parsers and LL[5] parsers are a few examples of to-down parsing.

1.4.2.2. Bottom-up parsing

Bottom-up parsing constructs a parsed tree from the leaves to the root, that is from bottom to top. The positive aspect of such systems is that it never suggests a tree that is not grounded to input but never reaches to the root, S. Grammar formalisms such as GPSG, HPSG, CCG, LFG, and DG are applied through bottom-up parsing. This study also follows a bottom-up parsing strategy for building a Tamil parser.

[5] LL- Left to right, Left most derivation type

LR[6] or shift reducing parsers like MALT[7] are some examples which follow bottom-up parsing.

1.4.3. Types of Parser

The parser is implemented in various ways with the suitable grammar formalism. Four major types of parsers, namely, rule-based parser, statistical parser, hybrid parser, and neural network-based parser are discussed below.

1.4.3.1. Rule-based Parser

Rule-based parsers use pre-written rules to describe the data. The main functor-argument relations are obtained using rule-based parsers. Grammar-driven dependency parsing is a type of rule-based parsing, which is formed from the combinations of context-free and constraint -based Dependency Parsing (DP). The natural language is formulated as formal language in rule-based parsers (Nivre, 2006). An input sentence of the language is validated, only when it is accepted by the grammar of the language, as the formal language is vital in this approach (Cf. Kübler, S., Ryan McDonald, Joakim Nivre and Graeme Hirst, 2009: 64-70). Weighted Constraint Dependency Grammar (WCDG) is another example of rule-based parsing (Krivanek, J., and Meurers, D., 2013).

Rule-based parsers are not widely used to parse Tamil sentences as all the available sentence structures cannot be accommodated in formal language using rules. Other problems include demands for more time and human resources; and dealing with complex sentences is a big challenge.

1.4.3.2. Statistical Parser

In Statistical parsers, grammar rules are associated with probability of a complete parse of a sentence. It is parsed by building treebanks. The commonly used grammar formalism in statistical parsing is Probabilistic Context-Free Grammars (PCFG). Data-driven dependency

[6] LR- Left to right, Right most derivation type
[7] MALT- MaltParser, developed by Johan Hall Jens Nilsson and Joakim Nivre at Växjö University and Uppsala University, Sweden is a system for data-driven dependency parsing, which can be used to induce a parsing model from treebank data and to parse new data using an induced model. (http://www.maltparser.org).

parsers (Nivre, 2006) follow a machine learning approach. Any sentence or phrase given as input is considered as a valid grammatical sentence and it is parsed.

Data-driven dependency parsing is sub-categorized into two types,

1. transition-based dependency parsing and
2. graph-based dependency parsing

MALT is the best example for statistical parser. However, statistical parser has drawbacks like

- New vocabulary is difficult to analyze
- New sentential constructions cannot be validated correctly. It can be improved by annotating bigger data (Jurafsky, 2000).

1.4.3.3. Hybrid Parser

A combination of rule-based and statistical parsing results in hybrid parsing, where the rules are applied to sentences after the machine learning. It gives better accuracy than the rule-based and probabilistic/stochastic models as both these models are inbuilt in this system. The requirement includes fully annotated treebank for probabilistic parsing and fully developed rules for the second phase of implementation (Cf. Kilian A. Foth, Wolfgang Menzel, 2006).

1.4.3.4. Neural Network-Based Parser

Neural network-based parsers refer to the application of neural network models. It works based on the dependency approach in both transition-based and graph-based dependency parsing. Yet, commonly found neural network parsers use transition-based dependency parsing, where the parser is powered by a neural network[8]. It has significantly improved the efficiency and accuracy of syntactic and semantic parsing.

The employment of a neural network classifier in a transition-based, greedy dependency parser, as suggested in a Stanford University research (1), is one illustration of this. This method works especially well at resolving the data sparsity problems typically encountered when training

[8] Neural Network is an information processing paradigm, which is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems (https://www.doc.ic.ac.uk).

transition-based parsers, and it does so without the requirement for intricate, hand-crafted features.

The input word embeddings are represented in vectors as shown in Figure (1.11).

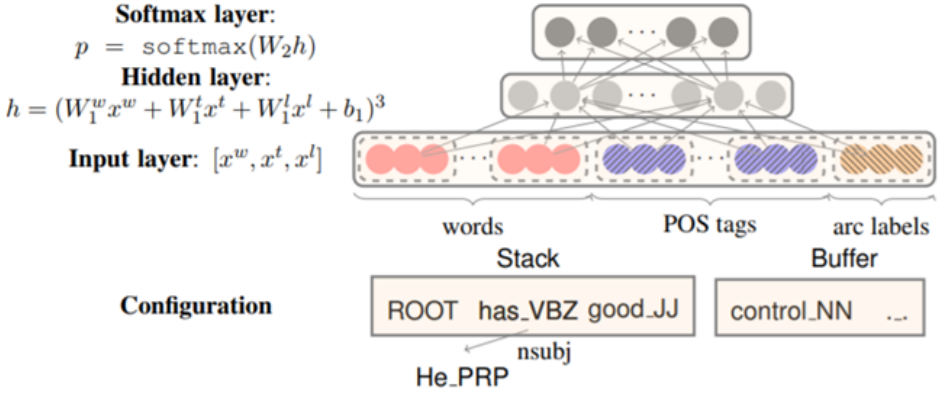


Figure 1.11: An example of Neural Network Schema (Danqi Chen and Christopher D. Manning, 2014)

1.4.4. Annotation Schema

Annotation schema is used in parsing to have a uniform pattern in marking features of a big data. There are many annotation schemas with tagset and guidelines are available in building parsers. In this section, the tagsets such as Penn tagset (Santorini, B., 1990), UCREL parsing tagset (ucrel.lancs.ac.uk), Prague tagset (ufal.mff.cuni.cz), Stanford tagset (De Marneffe, M. C., and Manning, C. D., 2008), Chinese Dependency tagset (Liu, H., and Huang, W., 2006), Anncorra tagset (Bharati, A., et al. 2009), Universal Dependency (UD) tagset (universaldependencies.org) are discussed. Among these tagsets, Anncorra tagset (Pāninian framework) and Universal Dependency tagset are taken into a detailed discussion, as it is mostly used in implementing parsers for Indian languages.

1.4.4.1. Penn Tagset

Penn Treebank (1989-1996), developed by University of Pennsylvania contains the POS tagged and syntactically bracketed forms of Brown corpus and Wall Street Journal. The Treebank has annotated 7 million words of POS tagged text, 3 million words of skeletally parsed text, over 2 million words of text parsed for predicate argument structure, and 1.6 million words of

transcribed spoken text annotated for speech disfluencies (Cf. Taylor, A., M. Marcus, and B. Santorini, 2003). It has 42 fine grained POS tags, 8 chunk tags, and 9 coarse grained relation tags, which are used in parsing. The major aim in introducing the tagset was to reduce the lexical and syntactic redundancy (Cf. Santorini, B., 1990).

1.4.4.2. UCREL Parsing Tagset

The UCREL tagset, developed by Lancaster University is used in semantic analysis systems of English. It has 21 coarse-grained discourse tags and 232 fine-grained semantic tags (Paul Rayson, et.al., 2004). The accuracy of the manually tagged system developed by Paul Rayson, Dawn Archer, Scott Piao and Tony McEnery had a precision of 91%.

1.4.4.3. Prague Dependency Tagset

The Prague Dependency Treebank was developed by the Prague School of Functional and Structural Linguistics. The project began in 1995 with the notion of following Praguian dependency tradition and building a Treebank similar to Penn Treebank. They have collected the database from the Czech National Corpus (Charles University, under the guidance of F. Čermák co-joint with other research centers/ institutions), and developed a three-layer system of tags: morphemic, syntactic at analytical level, and syntactic at tectogrammatical level (The Prague Dependency Treebank 3.0.). They had developed 68 fine-grained POS tagsets. (https://ufal.mff.cui.cz/).

1.4.4.4. Stanford Dependency Tagset

Stanford Dependency tagset was developed by a group of people from Linguistics and Computer Science as a part of an AI lab in 2005 for English. It was later extended to Chinese, Italian, Bulgarian and Portuguese. The main goal was to have a simple representation of the analyzed sentences which could be used by commons to extract word relations. The present Stanford Dependency Treebank has an approximate count of 50 relation tags (Cf. De Marneffe, M. C., and Manning, C. D., 2008). Most of these tags are also seen in the Universal Dependency tagset.

1.4.4.5. Chinese Dependency Tagset

Chinese Dependency Treebank 1.0 was released in May 2012 in Harbin Institute of Technologies Research Center for Social Computing and Information Retrieval (HIT-SCIR) for Mandarin Chinese, Chinese. It was developed by Wanxiang Che, Zhenghua Li, Ting Liu. It contains 49,996 Chinese sentences with 902,191 words, which were sourced from Peoples Daily newswire stories (1992-1996) and annotated with syntactic dependency structures. The data is provided in CoNLL-X format and in UTF-8 script. It has 13 word class tags and 34 fine grained dependency tags (Cf. Liu, H., and Huang, W., 2006).

1.4.4.6. Anncorra Tagset

Annotated Corpora (Anncorra) is developed based on the Pāninian Dependency grammar with *kāraka* and non-*kāraka* relations, aiming at a uniform representation of annotated corpus of Indian languages (Bharati, A., et.al, 2002). It is developed for parsing Hindi sentences and thus, the names of the tags used are in Sanskrit. Later, the same guidelines were adapted for other Indian languages (Marathi, Urdu, Bengali, Kannada, Telugu, Tamil, and Malayalam) (Cf. Tandon, J. and Sharma, D. M., 2017). It was even used by Amita, A. J. (2015) for English using the HyDT annotation scheme and hybrid approach (statistical + rule based) for parsing 2000 words.

The 19 fine-grained *kāraka* relations that are included in the Anncorra tagset are:

k1	karta ‘doer/agent/subject	k3	karana ‘instrument’ k4 sampradana ‘recipient’
pk1	prayojaka karta ‘causer’	k4a	anubhava karta ‘Experiencer’
jk1	prayojya karta ‘causee’	k5	apadana ‘source’
mk1	madhyastha karta‘mediator -causer’	k5prk	prakruti apadana ‘source

			material’
k1s	karta samanadhikarana- ‘noun complement of karta’	k7t	kAlAdhikarana ‘location in time’
k2	karma ‘object/patient’	k7p	deshadhikarana ‘location in space’
k2p	Goal Destination	k7	vishayadhikarana ‘location elsewhere’
k2g	secondary karma	k7a	according to and k*u sAdrishya ‘similarity/comparison’.
k2s	karma samanadhikarana ‘object complement’		

The 25 fine-grained non-*kāraka* relations include genitive case, adverbial and adjectival relations. It includes,

r6	shashthi ‘genitive/possessive’	rsp	address terms
r6-k1 r6-k2	karta or karma of a conjunct verb complex predicate	nmod__relc jjmod__relc rbmod__relc	relative clauses jo-vo constructions
r6v	kA ‘relation between a noun and a verb	nmod	participles etc. modifying nouns

adv	kriyAvisheSaNa 'manner adverbs'	vmod	verb modifier
sent-adv	Sentential Adverbs	jjmod	D-Rel modifiers of the adjectives
rd	direction	pof	part of units such as conjunct verbs
rh	hetu ‘reason’	ccof	co-ordination and sub-ordination
rt	tadarthya ‘purpose’	fragof	Fragment of
ras-k* upapada sahakArakatwa ‘associative’		enm	enumerator
ras-neg	Negation in Associative	rsym	ag for a symbol
rs	noun elaboration	psp_cl	relation between clause and postposition following that clause

1.4.4.7. Universal Dependency (UD) tagset

Universal Dependency is morphosyntactic annotation of languages, providing a good platform for developing multilingual parsers across domains and language typologies (http://universaldependencies.org). The notion of annotation scheme is extracted from Stanford dependencies, Google universal part-of-speech tags and the Interset interlingua for morphosyntactic tagsets in 2013 (McDonald et al., 2013). The parser works with only 3 modules:

1. Preprocessing (transliteration, sentence segmentation, tokenization)

2. M-layer annotation (positional tagging)
3. A- layer annotation (dependency annotation),

The fewer the number of modules, the wider number of language typologies can be covered. Indian languages like Hindi, Marathi, Sanskrit, Malayalam, Tamil, Telugu, and Urdu are included in the existing UD Treebank and Kannada and Pnar are upcoming languages listed in the table 1.1.

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

Figure 1.12: Syntactic tags used in Universal Dependency tagset
(http://universaldependencies.org)

Apart from these listed universal tags, there are 198 language-specific tags that are used in various language parsing systems. For instance, in Tamil, experiencer subjects in dative subject construction require a different tag as it is inflected with the non-nominative case and verbs do not agree with the so-called subject. The experiencer subjects are given with the tag ‘nsubj:nc’ i.e.. non-canonical subjects/dative subjects.

(1.22) *eṉa-kku* *paṭam* *pār.kk-a* *vēṇṭum*

I-DAT movie see-INF want

'I want to see a movie'

In the example (1.21), the dative-marked subject *eṉa-kku* 'I-DAT' is given with the tag 'nsubj:nc'

The table (1.1) provides the statistics of the Indian language in the UD project..

S.No.	Language	No. of annotated sentences	No. of UD tags	No. of language-specific tags
1	Hindi-HDTB	16,649	28	3
	Hindi-PUD	1000	38	9
2	Urdu	5130	16	1
3	Telugu	1328	42	11
4	Tamil-MWTT	690	38	17
	Tamil-TTB	600	31	4
5	Marathi	466	41	8
6	Sanskrit-UFAL	230	38	9
	Sanskrit-Vedic	3997	30	-
7	Bhojpuri	357	31	1

8	Malayalam	218	36	6
9	Kangri	288	24	1

Table (1.1): Statistics of Indian language in UD project
(https://universaldependencies.org)

1.5. Selection of Data

An enormous time was spent on carefully choosing the corpus as the sentences in the corpus should be exhibiting various kinds of constructions. Domain based corpus was finally chosen for study as each domain exhibited certain types of constructions. The notion of the text 'domain' has been seen as a major constraint on the applicability of knowledge. Domain-based parsing is a parsing technique that exploits knowledge about domain-specific properties of terms in order to determine "optimal" parse trees for natural language sentences. (Sekine, S., 1997). There are domain dependencies on syntactic distribution. Since most operations operate within a specific domain, application of domain-specific treebank will help in improving the accuracy. Parsing performance is the best when the test data is from the same domain and it is worse when the data is extracted across the domains. It may not be useful to use a different domain corpus even if the size of the corpus is too large.

In the initial task, conversational components were carefully avoided as the morph cannot recognise dialectal words and tackling ellipses would be very challenging. As an extended study, some corpus on social media and speech conversation was added to study the challenges posed by such data.

The domain-specific data with 1000 sentences from each domain was selected from various resources. The specific domains include, tourism, sports, agriculture, social media and speech conversation, which has a variety of sentences as described below:

S.No.	Sentence constructions	No. of occurrences (out 0f 1000 sentences)				
		Tourism	Agri -culture	Sports	Social media	Speech Conver -sation
1	Relative clause constructions	148	211	389	340	332
2	Clausal subjects	56	14	23	9	11
3	Infinitive constructions	226	187	170	121	38
4	Clausal complements	122	42	344	50	389
5	Imperative constructions	8	171	2	19	135
6	Passive constructions	164	81	15	93	35
7	Copula constructions	226	279	154	103	17
8	Other constructions	132	89	125	140	37

Table 1.2: Types of sentence constructions in each domain

1.5.1. Tourism

The tourism data is extracted from Tamil Nadu tourism website which has simple, complex and compound sentences. The following constructions are found in the tourism data.

(i) Copula constructions with or without copula is a common occurrence in the data.
Example (1.23) *itu varalāṟṟu ciṟappuvāynta taḻam* 'This **is** a famous historical spot'
Example (1.24) *itu ōr paruvakkāla nīrvīḻcciy**ākum*** 'This **is** a seasonal waterfall'

(ii) Names of places, people natural resources and wildlife are very frequent occurrences in the data
Example (1.25) ***kāvēri** āṟu tāṉ **tamiḻnāṭṭil** pala māvaṭṭaṅḻiṉ nīr ātāram*
'Cauvery river is the water source for many districts in **TamilNadu**'

(iii) Historical reports in passive constructions
Example (1.26) *tañcay periya kōvil 2023il **putuppikkappaṭṭatu***

‘Tanjore temple was **renovated** in 2023’

1.5.2. Sports

Sports data is scrolled from Vikatan sports magazine, which has simple, complex and compound constructions. The sentences are mostly in active voice and are filled with a lot of proper nouns. Since the data is taken from the Vikatan magazine report, the sentences are reported in past tense.

(i) Names of person and sports are very common occurrences

Example (1.27) ***kultīp*** *5 vikkeṭṭuka̲̲l̲ay vīl̲ttiṉār*

‘Kuldeep took 5 wickets’

(ii) Occurrence of English terms related to each game is seen which are marked foreign in morph

Example (1.28) *vārṉē* ***ṭak avuṭ*** *āṉār*

‘Warne got **duck out**’

1.5.3. Agriculture

Agriculture data has a number of biological names of flora and fauna, names of diseases, insecticides/pesticides/fertilizers and nutritional/biological facts, which is extracted from the Tamil Nadu government’s Ministry of Agriculture website. Passive and copula constructions are very prevalent.

(i) names of flora, fauna and diseases

Example (1.29) *nel vakayka̲̲l̲il* ***oraycā cattayvā*** *oru vakay*

‘Among rice varieties, **Oryza Sativa** is one such variety’

(ii) Sentences stating nutritional, historical or biological facts

Example (1.30) *itu te̲̲r̲kāciyāvil tōṉ̲̲r̲iyatu*

‘It is originated in South Asia’

1.5.4. Social Media

Movie reviews from facebook and twitter were taken as the social media text. The text had many smileys, hash tags, at the rate of symbols, etc,. The data looked code-mixed and thus, a lot of English words were found in the data. Such words are marked 'foreign' in the morph. Short forms of many words were used. English words were typed in Tamil and vice-versa was seen, Dialectal words were also found in the data.

(i) Symbols and smileys
Example (1.31) *makiḻcci* 🥲 'happy 🙂'
Example (1.32) ***@***ARRahman '@ A.R. Rahman'

(ii) Code-mixed data
Example (1.33) *50* ***to*** *60* ***audience*** *iruntārkaḷ* '50 to 60 audience were there'

1.5.5. Speech conversation

Speech by Dr. Sivaraman was taken to be the content of speech conversation data, extracted from https://www.youtube.com/@tamilspeechbox. Speech conversation data is the most complex domain to annotate as it has dialectal variations and it is filled with ellipses. The dialectal vocabulary that he uses in his speeches are mostly found in Tirunelveli district of Tamil Nadu. Since Tamil is a diglossic language, speech data will vary with written Tamil. Repetition of words or clauses is seen. Extra words that are not grammatically related to the sentence are used for which the relation 'dep' is marked.

(i) Some words that are not grammatically related
Example (1.34) *avarkaḷ* ***vantu*** *varuvārkaḷ* 'They will come'

vantu is an extra word which is not grammatically related. It is used in casual conversations. Such relations are marked 'dep' to the root of the sentence.

(ii) Diglossic words
Example (1.35) *nāṉ* ***varēṉ*** 'I am coming'

Here, ***varēṉ*** is the diglossic word for ***varukiṟēṉ***

1.6 Methodology

This section discusses the methodology implemented for the study in detail. Corpus collection, different layers of annotation, theoretical background and its implementation technique, domain adaptation of existing parser are the different aspects of methodology implemented in this study. A step-by-step methodology of the thesis is discussed below in detail:

1.6.1 Corpus collection

The domain- specific corpus with 1000 sentences from each domain was carefully collected. Various domains include tourism, agriculture, sports, social media and speech conversation. The corpus had a variety of sentences including simple, complex and compound. Since the data was scrolled from various sources, the data was raw and had to be cleaned. The sentences were changed to '.txt' format for further cleaning.

Cleaning of corpus included removing unwanted elements like photos, website URLs, spacing issues, advertisement contents, etc. The spelling errors were left untouched and other errors like ungrammatical sentences were removed. The sentences were initially tokenized, followed by word tokenization.

1.6.2 Treebank pre-processing and dependency relations

Treebank pre-processing includes multi-word tokenization, POS tagging, and morph tagging. Multi-token words are identified and it is decided to split them if the two words syntactically belong to two different categories. The tagsets for these are described in detail in chapter 3. Following the pre-processing, fine-grained dependency relations are marked for the words in the data, which is described in chapter 4.

1.6.3 Training the model and error analysis

The model is trained using the trankit model of parsing. It is done over the existing parser, trained at IIIT-H. Tamil domain-specific sentences were given as the input. Join-Token and

Sentence Splitter would tokenize sentences and words. Multi-word Token Expander identifies the multi-word token and splits into two different words if they are syntactically different categories. Joint Model for POS, Morphological Tagging provides POS and morphological tags to the data. The Dependency Parsing module provides syntactic tags and thus, the tree is parsed.

1.6.4 Fine-Tuning

Fine-tuning in Natural Language processing refers to the re-training of data which is already pre-trained for some other data. The re-training of domain-specific data from the already existing Trankit model developed by IIIT-Hyderabad was done for the thesis. Accuracy is calculated for each domain and error analysis is done to improve the accuracy.

1.6.5 Arguments for linguistic and computational methodology

Dependency grammar is the linguistic methodology used as dependency grammar marks relations by head-dependent relations as opposed to constituency parsing which generates trees as constituents/ phrases. Constituency grammar fails with free-word order language as the order of constituents keep changing.

Annotation schema used is Universal Dependencies. It is chosen over other schemas, especially AnnCorra as UD's approach is purely syntactic and AnnCorra is syntactico-semantic.

Data-driven approach is the computational methodology used as the results for domain-specific data is better with data- driven approach. As one of the domains is speech conversations, rule-based parsing will not work.

1.7 Organization of the thesis

The thesis is organized into 6 chapters

Chapter 1: Introduction

Introduction deals with the introduction of parsing, kinds of parsers, grammatical structure, kinds of available grammatical frameworks, kinds of available treebanks, the structure of Tamil, selection of data, and methodology of this work as seen before.

Chapter 2: Universal Dependency Parsing: A Review

A review of all papers, articles and theses published on Universal Dependency parsing in global, Indian, Dravidian languages with special reference to Tamil. A review on domain specific parsing papers with special reference to tourism, sports, agriculture, social media and speech conversation domains.

Chapter 3: Treebank Pre-processing

Pre-processing includes copus cleaning, tokenization, multi-token word identification, and developing Morph and POS framework with language specific examples. The same morph and POS framework was used as guidelines for developing IIIT Tamil Syntactic Parser. This thesis has adapted the same existing Tamil Syntactic Parser developed at IIIT- Hyderabad for studying the domain-specific constructions.

Chapter 4: Domain-Specific Syntactic Treebank

A set of syntactic treebank frameworks for Tamil are presented for all the tags used in the treebank. The framework is explained with Tamil examples using dependency tree diagrams. Language specific tags are also included in this chapter, which was a contribution to MWTT and IIIT, Hyderabad Tamil Syntactic parser and the same was used to do an extended study on domain - specific constructions.

Chapter 5: Evaluation and error analysis

This chapter discusses the process of fine-tuning the data; about various parser evaluation schema and how it is evaluated. This chapter also evaluates the parser domain wise and illustrates the statistics of the POS and syntactic tags used in each domain. Later, error analysis is done to improve the accuracy of the parser.

Chapter 6: Conclusion

Overall remarks on parsing, worked strategies, significance of this research and the future works that can be done are discussed in this concluding chapter.

Chapter 2

Universal Dependency Parsing: A Review

2.1 Universal Dependency Parsing

The official details of Universal Dependency parsing are recorded in the website, https://universaldependencies.org/. The website has a record of number of working languages; general morphological, POS, and dependency guidelines for all languages and possible extensions for language specific features; number of treebanks in each language; number of features, POS tags, and dependency relations used in each language; statistical calculations of tags used; and a record of upcoming languages too.

Several articles, theses and books have been written on Universal Dependency parsing. This section lists down all such works.

- Paper titled 'Universal Dependencies: A cross-linguistic typology' (de Marneffe,et al., 2016) adds to offering a cross-linguistic typology of universal dependencies, which also sheds light on the typology that underlies the Stanford Dependencies representation.
- 'Universal Dependency Parsing from Scratch' by Qi,P., et al. (2019) described CoNLL 2018 shared task where a neural pipeline was used and it out performed the state of the art results and accuracy was improved by 0.11 % in overall F1 score.
- A conference paper on 'Multilingual Dependency Parsing from Universal Dependencies to Sesame Street' by Joakim Nivre (2020) discusses the successful five years of advancements implemented in UD
- A detailed introduction to Universal Dependencies by Marneffe, et al. in 2021 acts as a referral guide for UD grammar.

Parsers are being implemented worldwide for various languages. This section deals with a review of parsing in the following languages:

(i) World languages

(ii) Indian languages

(iii) Tamil

2.2 Parsing in world languages

A parsing algorithm recorded to be the earliest was proposed by Yngve (1955). Yet, most of the parsers were developed in the early 1990s. Such implemented parsers for world languages include:

- Collins's (1999) statistical parser for Czech using Prague Dependency Treebank
- Eugene Charniak's (2000) maximum-entropy parser for English
- Bikel and Chiang's (2000 first statistical model on Chinese Treebank
- DeSR, developed by Yamada and Matsumoto (2003) for English
- Dubey and Keller's (2003) proposal of a probabilistic parsing for German
- Stanford parser (2003), a statistical parser, using lexicalized PCFG (Probabilistic Context-Free Grammar), developed by Dan Kleinbeing for English and further extended to Arabic, Chinese, French, German and Spanish
- A probabilistic parser with supervised learning based on PCFG for English (Collins, 1997)
- Robust Accurate Statistical Parsing (RASP) System, a hybrid domain independent English parser (Cf. Briscoe, et.al, 2006)
- MALT (2007) and MST (2005) (developed by Johan Hall, Jens Nilsson and Joakim Nivre at Växjö University and Uppsala University, Sweden), a transition based parser
- ISBN (Incremental Sigmoid Belief Networks), a trainable dependency parser (Cf. Titov, I. and Henderson, J., 2010)
- Carnegie-Mellon's Link Grammar parser, built for English, Arabic, Russian and Persian
- Seraji, M., Jahani, C., Megyesi, B., and Nivre's (2014) work on Persian by obtaining the data from large-FARSDAT
- Universal Dependencies (UD) (McDonald et al., 2013), a project developed by Joakim Nivre, which is involved in developing a cross-linguistic study, maintaining a treebank annotation for 60 languages with 102 treebanks

- UD Parser for Persian was developed with 6000 sentences with an average of 25 word length in each sentence (Seraji, et al., 2016)

2.3 Parsers in Indian languages

Parsing is one such area, which has to be explored in depth for Indian languages. Some of the Indian languages including Hindi, Urdu, Telugu, Kannada, Tamil, Bengali, Marathi, and Assamese have delved into areas of parsing, which are still work in progress. (Cf. Monika T. Makwana and Deepak C. Vegda, 2015). In fact, Tandon, J., and Sharma, D. M. (2017) has come up with a unified strategy for parsing Indian languages using the Pāninian framework. Research related to cost-effective methods of building dependency parser for Indian languages are also in the current trend (Cf. Tammewar, A. 2015). The list of implemented parsers in Indian languages is discussed below:

- Nivre (2009) optimized MALT parser for Hindi, Bengali and Telugu. With coarse-grained tagset, the respective accuracies are 81.1%, 79.6% and 63%. But, when a fine-grained tagset is used, it has lower accuracies, i.e. 75.3%, 72.9% and 58.5%.
- Hindi, Bengali and Telugu sentences are tested with MALT and MST (data-driven parsers) by Bharat Ram Ambati, et.al. (2009), where MALT has a better performance than MST. The report has a final average score as 88.43%, 71.71% and 73.81% respectively.
- A bidirectional dependency parser for Hindi, Bengali and Telugu is proposed by Prashanth Mannem (2009), which shows the accuracy of 71.63%, 59.86% and 67.74% respectively when run with test data. The same data has better accuracies with the coarse-grained tagset, 76.90%, 70.34% and 65.01% respectively.
- A constraint based dependency parsing system for Bengali with Pāninian Grammar formalism is proposed by Sankar De, et.al. (2009), which is trained with 1000 annotated sentences, and evaluated with 150 sentences. It has the accuracy of 79.81%, 90.32% and 81.27% for labelled attachments (LAS), unlabelled attachments (UAS) and label scores (LS) respectively.
- Aniruddha Ghosh, et.al. (2009) trains Bengali data using CRF and was implemented using a rule-based algorithm. It results in 74.09% (LAS), 53.09% (UAS) and 61.71% (LS).

- Sanjay, et.al. (2009) has run Bengali sentences on a data-driven parser and hybrid parser. The wrongly annotated sentences are given rules to improve the accuracy. A special look at subject, object, location and relation is observed.
- Rahman, Mirzanur, et.al. (2009) analyse the issues in areas of parsing Assamese sentences when tagged with 7 tags based on CFG formalism. Later, rules are developed accordingly and algorithms are modified from Earley's Algorithm to solve those issues.
- A constraint-based Hindi dependency parsing system with the accuracy of 62.20% (LAS) and 85.55% (UAS) is implemented by Meher Vijay Yeleti and Kalyan Deepak (2009).
- Bharat Ram Ambati, et.al. (2010) analyse the role of linguistic features in data-driven dependency parsing for Hindi and found that accuracy gain is seen when adding morphosyntactic features like case and TAM features. They had finally gained 2% accuracy (76.5% in total) after combining morph features from two different parsers.
- Antony P.J. (2010) has developed a statistical syntactic Kannada parser using Penn Treebank with 1000 POS tagged sentences using SVM POS tagger. It is implemented using supervised machine learning and is evaluated using SVM algorithms. As a result, they claim to have good accuracy.
- B.M.Sagar (2010) has developed a CFG for Kannada parser and finally proposes that top-down parser is best suited for Kannada.
- Navanath Saharia, et.al. (2011) have used CFG to parse the simple sentences of Assamese, which is not implemented.
- B.Venkata S. Kumari, et.al. (2012) use a combination of MALT and MST parsers which shows LAS 90.66% for gold standard and 80.77% for automatic tracks.
- Karan Singh, et.al. (2012) propose a two-stage approach for Hindi Dependency Parsing using MALT parser. Their system has a record of 90.99% (LAS) for the gold standard.
- Uma Maheshwar Rao G., K. Rajya Rama, A. Srinivas (2012) has worked on the Dative case towards building a parser. Various functions of the dative marker are discussed and a flowchart is developed to build a robust parser for Telugu.
- Sambhav Jain, et.al. (2013) has added the ontological features to Hindi dependency parser which added the accuracy improvement of 1.1% (LAS) for 1000 sentences and 0.2% (LAS) for 13371 sentences.

- A Lexicon parser for Devanagiri script (Hindi) is developed by Swati Ramteke, et.al. (2014), which generated semantic parsed trees with an accuracy of 89.33% when run with unambiguous sentences. Rule-based approach was used to resolve the lexical ambiguities.
- Arpita Batra, Soma Paul, and Amba Kulkarni (2014) had worked on the constituency analysis for Hindi using four approaches. Adjacency global, adjacency greedy, dependency global and dependency greedy were applied for 2322 sequences of words. Applying all these approaches, 92.85% (using global dependency algorithm and syntactic rules) accuracy was obtained.
- Dhanashree Kulkarni, et.al. (2014) has taken up CFG as the grammar formalism and used the same in Top-Bottom and Bottom-Top parser for Marathi. The final outcome of the paper was to develop (computerized) grammar checking for Marathi text from CFG perspective.
- A Combinatory Categorical Grammar (CCG) Telugu treebank is created using CCG lexicon and dependency Treebank and it is tagged with CCG supertags as features to Telugu dependency parser. An improvement of 1.8% in UAS and 2.2% in LAS (especially on verbal arguments) was observed when implemented using MST parser (Cf. Kumari, B. and Rao, R. R., 2015).
- Telugu Dependency parser, developed by Nagaraju, G. et.al. (2016) have used bottom-up parser and parsed 200 Telugu sentences using *kāraka* relations. Out of 200 sentences, they have obtained 178 correct parsed sentences. As a whole, 880 words were correctly tagged and 140 were incorrect and thus, they claim the precision to be 99.
- 'Improving Transition-Based Dependency Parsing of Hindi and Urdu by Modeling Syntactically Relevant Phenomena', by Bhat, R. A., Bhat, I. A., and Sharma, D. M. (2017) have used *kāraka* and non-*kāraka* relations and annotated the inter-chunk dependencies manually. They have implemented a transition-based dependency parser with syntactic features and obtained an accuracy of 87.82% in the trained set and 87.72 in test data of LAS.
- Dependency parser for Sanskrit verses by Amba Kulkarni and her team in 2019 achieved parsing 150 sentences, leaving behind 45 sentences in a total of 195 sentences.
- Kulkarni, A. has developed a Sanskrit Parser (2021) and published a book titled 'Sanskrit Parsing: Based on the Theories of Śābdabodha'.
- P.Sangeetha (2022) has developed a rule-based dependency Telugu parser using Pāninian framework and has come up with an accuracy of UAS 90.3% and LAS 84.1%

2.4 UD parsing in Indian languages

- 'Universal Dependency Parsing for Hindi-English Code-Switching' by Bhat,et al., in 2018 states the challenges faced in annotating code-switching data (tweets) and the possible ways to overcome the same. Their results were 1.5% LAS points better than annotated data when neural networking was used and 3.8% LAS points improved while decoding the code-switching data.
- Bhojpuri UD, developed by Atul Kr. Ojha and Daniel Zeman have attained 79.69% UPOS accuracy and 77.64% XPOS accuracy.
- A manually tagged Odia corpus of 100 sentences using UD guidelines produced an accuracy of 42.04% UAS and 21.34% LAS (Parida, S, et al. 2022)

2.5 Parsers in Tamil

- Tamil, belonging to the Dravidian language family, is morphologically rich. It has a (S)OV word order with agglutinative morphology. Hence, building a parser for Tamil is a challenging task. This section lists the Tamil parsers with grammar formalisms and techniques used in their respective parsers.
- Hybrid approach combining PSG and DG with Lexicalized and Statistical Parsing (LSP) is used by Selvam, M., Natarajan, A. M. and Thangarajan, R. (2008) with 500 tags and 31 dependency relations on Tamil. 3261 sentences with 51026 words are used and as a result, 73% accuracy in trained data and 65% accuracy in test data with just 600 trained sentences were obtained. The lacuna is seen in their choice of their tagset which had 500 tags.
- A Tamil syntactic parser, proposed by K. Sureka, Dr. K. G. Srinivasagan and S. Suganth (2014) works on dependency grammar and follows a hybrid approach, with clause boundary identifier. After adding the module, the result obtained is that out of 150 sentences, 120 are parsed correctly.
- Vigneshwaran (2017) has worked on Tamil parsing based on cognitive grammar as the theoretical grammar and Pāninian framework as the computational grammar. The main argument revolves around parsing Tamil sentences at discourse level, as it claims that sentential analysis is not enough to get an idea of the complete context of the text.

- Vijay Sundar Ram and Sobha Lalitha Devi have developed a dependency parser for Tamil using multiple formalisms: UD and AnnCorra. MALT parser was used for implementation and resulted in an accuracy of UAS 79.27% and 73.64 LAS.

2.6 UD parsing in Tamil

- Loganathan Ramasamy and Zdeněk Žabokrtský (2011) have done initial experiments with Tamil dependency parsing using rule-based approach (with an accuracy of 79% (LAS) in the trained data and 61% with the test data) and corpus based approach (with an accuracy of 75%. Finally, it is concluded that both the approaches have failed in identifying coordination nodes.
- Universal Dependency has extended its system to Tamil, by developing a Tamil Treebank (from Prague dependency Treebank) (Cf. Ramasamy, Loganathan and Zdeněk Žabokrtský, 2012), which has universal tagsets and just involves three processes: Pre-processing (transliteration, sentence segmentation, and tokenization); M-layer annotation (positional tagging) and A- layer annotation (dependency annotation) with 217 distinct tags (including all 9 positions). 96% of the test data was unambiguous; 3% was ambiguous with 2 tags and tokens with 3-4 tags were just 1% which is negligible. Altogether, 21 dependency relations are used for labeling edges. It has an accuracy of 69% when trained with 690 sentences.
- Sankaravelayuthan, R., et al. have developed a Tamil dependency parser in 2019 based on the Stanford dependency model for tourism domain. The model could answer 50,000 questions in the tourism domain is their claim.
- K. Sarveswaran and Gihan Dias developed *ThamiZhi*UDp, a neural based dependency parser for Tamil in 2020, following UD pipeline and formalism. It has achieved an F1 score of 93.27 and LAS 62.39.
- K. Sarveswaran (2022) has developed a grammar based deep syntactic parser using LFG and trained the model using the UD framework.
- K.Parameswari and her team in 2024 released a Tamil Syntactic Parser, developed at IIIT, Hyderabad using UD framework with an accuracy of UAS 87.4 and LAS 79.6.

2.7 Domain-specific adaptation of parsers

- A Domain-adapted Dependency Parser for German Clinical Text by Kara, E, et.alin 2018 did a study on clinical texts and the accuracy of the parser improved from 42.15 to 78.26 when the same domain corpus was tested.
- METAL parser is used primarily for German text with subject-encoded dictionaries with a maximum accuracy of 80%.
- Cico, a simple parser using domain-based parsing, is particularly well suited for parsing natural language sentences of technical nature.
- A paper on Fine-Tuning for Domain Adaptation in NLP was done for Italian works well for generic text but not for other domain-specific data.

Chapter 3

Treebank Pre-processing

3.1 Morphology of Tamil

Tamil has an agglutinative morphology. It has verb conjugation, case marked nouns and noun declension. Compounding of nouns and verbs, reduplications and complex system of tense and aspect in Tamil poses a big challenge to work with Tamil morphology. From a computational point of view, creating word paradigms of nouns, verbs, and all the other grammatical categories help the system learn the lexical words and its derivations.

3.2 Morphological tags

Universal POS features are used to define morphological features for Tamil. Those morphological features and its values are listed in the table (3.1).

	Features	**Values**
pronominal type	PronType	personal (Prs), reciprocal (Rcp), article (Art), interrogative (Int), relative (Rel), exclamative determiner (Exc), demonstrative (Dem), total (Tot), indefinite (Ind)
numeral Type	NumType	cardinal (Card), ordinal (Ord), fraction (Frac)
possessive	Poss	Yes
reflexive	Reflex	Yes
foreign word	Foreign	Yes
abbreviation	Abbr	Yes
wrong spelling	Typo	Yes
gender	Gender	masculine (Masc), feminine (Fem), neuter (Neut)

animacy	Animacy	animate (Anim), human (Hum), inanimate (Inan)
number	Number	singular (Sing), plural (Plur), dual number (Dual), trial number (Tri), collective (Coll)
case	Case	nominative (Nom), accusative (Acc), instrumental (Ins), dative (Dat), ablative (Abl), allative (All), benefactive (Ben), comitive (Com), locative (Loc), genitive (Gen), vocative (Voc)
definiteness/ state	Definite	definite (Def), indefinite (Ind)
Verbal form	VerbForm	finite (Fin), infinite (Inf), participle (Part), gerund (Ger), verbalnoun (Vnoun)
mood	Mood	indicative (Ind), imperative (Imp), conditional (Cnd), potential (Pot), desiderative (Des), necessity (Nec)
tense	Tense	present (Pres), past (Past), future (Fut)
aspect	Aspect	progressive (Prog), perfective (Perf)
voice	Voice	active (Act), passive (Pass), causative (Cau)
polarity	Polarity	positive (Pos), negative (Neg)
person	Person	1 2 3
polite	Polite	formal (Form)
clusivity	Clusivity	inclusive (In), exclusive (Ex)

Table (3.1): Morphological guidelines for Tamil in UD framework

3.2.1 PronType: pronominal type

Pronominal type is usually related to pronouns. In Tamil, personal, reciprocal, interrogative, relative, demonstrative, negation, and indefinite are the types found. Each type is described in detail with examples below:

3.2.1.1 Prs: personal or possessive personal pronoun or determiner

Personal pronouns are alternatives to proper nouns. It includes possessive personal pronouns as well in Tamil. The list includes *nāṉ* 'I', *nī* 'You', *avaṉ* 'He', *avaḷ* 'She', *avar* 'He/she (hon)', *atu* 'It', *nāṅkaḷ* 'We', *nīṅkaḷ* 'You (hon)', *avarkaḷ* 'They', *avay* 'Those', *nām* 'We', *uṉ* 'Your', *uṅkaḷ* 'Your (pl/hon)', *avarkaḷuṭayya* 'Theirs', *eṉ* 'My', *eṉṉuṭayya* 'Mine', *avaḷuṭayya* 'Hers', *avaṉuṭayya* 'His', *ataṉuṭayya* 'It's', *uṉṉuṭayya* 'Yours', *uṅkaḷuṭayya* 'Yours (pl/hon)', *nammuṭayya* 'Ours'.

Example (3.1) ***uṅkaḷ*** *vīṭu* 'Your(hon) house'
Example (3.2) ***eṉ*** *puttakam* 'My book'

If reflexive feature is found in the personal pronoun, then the feature value will be written as (PronType= Prs| Reflex= Yes)

Example (3.3)
avaḷayyavaḷēp *pārttukkoṇṭāḷ*
'She saw **herself'**

3.2.1.2 Rcp: reciprocal pronoun

Reciprocal pronouns have plural subjects. The actions done by every member of the group are explained to every other member of the group by the predicate.

Example (3.4)
oruvarayyoruvar *taḻuvikkoṇṭaṉar*
'They hugged **each other'**

3.2.1.3 Art: article

Articles are determiners that specify the definiteness of nouns. In Tamil, the usage is minimal when compared to English.

Example (3.5)
oru *nāṟkāli*
'**A** chair'

In example (3.5), *oru* 'a' refers to an indefinite determiner. Definite determiners are a part of personal pronouns or demonstratives in Tamil.

3.2.1.4 Int: interrogative pronoun, determiner, numeral or adverb

Pronouns replacing nouns in the question are interrogative pronouns. Interrogative pronouns and adverbs are seen in Tamil. *yār* 'who'and *eṉṉa* 'what' are pronouns and *ēṉ* 'why'*, eppaṭi* 'how'*, etaṟku* 'for what'*, eṅkē* 'where'*, etaṉāl* 'why'*, ētu* 'how' are tagged adjectives.

Example (3.6)
yār avar?
'Who is he?'

Example (3.7)
avaṉ ***eṅkē****?*
'**Where** is he?'

3.2.1.5 Rel: relative pronoun, determiner, numeral or adverb

Pronoun which introduces a relative clause to give more details about the preceding noun/NP is a relative pronoun. These are interrogative words but the usage is relative.

Example (3.8)
avaṉ ***yāruṭaṉ*** *vantāṉō avaṉōṭē pōkalām*
'He can go with **whom** he had come'

3.2.1.6 Exc: exclamative determiner

Speaker's expression of exclamation on modified nouns is indicated by a word like *eṉṉa* 'what' in Tamil. Such exclamation is tagged determiner.

Example (3.9)
eṉṉa *oru makiḻcci!*
'**What** a surprise'

3.2.1.7 Dem: demonstrative pronoun, determiner, numeral or adverb

Demonstratives are either determiners or adverbs in Tamil. They indicate the entities that are referred to. It is also used to highlight an entity separately. Words like *anta* 'that' and *inta* 'this' are determiners and *aṅke* 'there', *iṅkē* 'here', *appōtu* 'then', *ippōtu* 'now' are adverbs.

Example (3.10)
anta *payyaṉ*
'That boy'

Example (3.11)
iṅkē *vantāṉ*
'Came **here'**

3.2.1.8 Tot: total (collective) pronoun, determiner or adverb

In Tamil, collective or totality meaning is expressed by the determiners and adverbs.

Example (3.12)
Determiner:
aṉayttu *puttakaṅkaḷum uḷḷaṉa*
'All the books are there'

Example (3.13)
Adverb:

malar ***eppoḷutum*** *paṭippāḷ*

‘Malar **always** studies’

3.2.1.9 Ind: indefinite pronoun, determiner, numeral or adverb

Some numerals are found in Tamil, which expresses indefiniteness.

Example (3.14)

raku ***cila*** *pommaykaḷay vāṅkiṉāṉ*

‘Raghu bought **some** toys’

3.2.2 NumType: numeral type

Numeral type is a complex system in many languages. It is quite simple in Tamil. It has cardinal, ordinal, and fraction as sub-types. From the syntactic point of view, some numtypes fall into adjectives. Others remain as numbers or determiners.

3.2.2.1 Card: cardinal number

Numbers in base form without any change, used for counting are cardinal numbers. It is classified under NUM in POS.

Examples (3.15 and 3.16)

3.15 ***iraṇṭu*** *nāṭkaḷ* ‘Two days’

3.16 ***mūṉṟu*** *muṟay* ‘Three times’

3.2.2.2 Ord: ordinal number

Ordinal numbers are classified under ADJ in Tamil. Ordinal numbers represent the position or rank of an object or a person.

Examples (3.17 and 3.18)

3.17 ***iraṇṭāvatu*** *aṟay* ‘Second room’

3.18 ***pattāvatu*** *iṭam* ‘Tenth place’

3.2.2.3 Frac: fraction

Fraction denotes a part of a whole number. It is said in a word in Tamil. It is a sub-type of cardinal numbers.

Examples (3.19 and 3.20)
3.19 ***aray*** *āppiḷ* **'Half** apple'
3.20 ***mukkāl*** *ēkrā* '**Three-fourth** of an acre'

3.2.3 Poss: possessive

If the word has possessiveness information encoded in it, then it is marked as poss=Yes.
Pronouns and nouns carry this information in Tamil.

Example (3.21 and 3.22)
3.21 ***ta̲n̲atu*** *puttakam* 'his/her book'
3.22 ***ava̲l̲atu*** *puttakam* 'her book'

3.2.4 Reflex: reflexive

If the word has reflexivity in it, then the word is marked with Reflex=Yes. Pronouns are encoded with reflexivity information in it.

Example (3.23)
ava̲l̲ ***ta̲n̲n̲ayttā̲n̲ē*** *kaɳɳāṭiyil pārtā̲l̲*
'She saw **herself** in the mirror'

3.2.5 Foreign: is this a foreign word?

If any foreign word which is not a loan word or foreign name appears on the data, then the word is marked with foreign=Yes.
Example (3.24)
*malar "**tomorrow sunday holiday**" en̲ru kūr̲iṇā̲l̲*
'Malar said, tomorrow is sunday, holiday'

In the above example, highlighted words are English words which are neither loan words nor foreign names.

3.2.6 Abbr: abbreviation

Abbreviations are shortened forms of a word or one whole phrase. It is usually seen in nouns and proper nouns. Such abbreviated words are marked with the feature, Abbr=Yes.

Example (3.25-3.30)
3.25 ***aynā***- *aykkiya nāṭu* 'United Nations'
3.26 ***tanā***- *tamiḻ nāṭu* 'Tamil Nadu'
3.27 ***mī***.- *mīṭṭar* 'metre'
3.28 ***ki.mu***.- *kiṟistuvukku muṉ* 'Before Christ'
3.29 ***ki.pi.***- *kiṟistuvukku piṉ* 'After Christ'
3.30 ***ki.mī***.- *kilō mīṭṭar* 'kilometer'

3.2.7 Typo: is this a misspelled word?

Typo are typographical errors that occur mainly due to unusual character coding. In Tamil, misspelled words are marked Typo=Yes.

Example (3.31)
aṟuvi 'waterfall' Typo=Yes
(aruvi is the right spelling)

Typo=Yes is also used when the word is wrongly split. An addition of the tag 'goes with' is marked along the second word which goes with the first split word.

Example (3.32)
1 ***nīrpp*** Typo=Yes
2 ***pūcaṇi*** goes with 1

Here, *nīrpppūcaṇi* is one word meaning pumpkin. It has undergone a wrong split and thus, tagged ‘goeswith’ with the first word.

3.2.8 Gender: gender

Gender is usually a lexical feature of nouns and derived nouns, and an inflectional feature of verbs and pronouns. It has an agreement with nouns.

3.2.8.1 Masc: masculine

Masculine gender denotes males in general.

Examples (3.33-3.35)
3.33 ***payyaṉ*** ‘boy’
3.34 ***rāmu*** ‘Ram’
3.35 ***āṇkaḷ*** ‘Gents’

3.2.8.2 Fem: feminine

Feminine gender denotes females in general.

Examples (3.36-3.38)
3.36 ***peṇ*** ‘Woman/Lady’
3.37 ***cītā*** ‘Sita’
3.38 ***makaḷir*** ‘Women’

3.2.8.3 Neut: neuter

Nouns that refer to common gender or that belong to neither masculine nor feminine are categorized as neuter.

Examples (3.39-3.42)
3.39 ***kuḻantay*** ‘baby’
3.40 ***puttakam*** ‘book’
3.41 ***puli*** ‘tiger’

3.42 ***mayil*** ‘peacock/peahen’

3.2.9 Animacy: animacy

The term ‘Animacy’ refers to the state of being alive. In linguistics, it is a lexical feature of nouns and derived nouns (that are alive) and an inflectional feature of other categories like verbs and pronouns. It has an agreement with the nouns.

3.2.9.1 Anim: animate

Animate nouns include human beings, animals and birds, fictional characters like unicorns and fairy, names of professions like teacher, doctor, etc. Personified inanimate nouns are also marked animate.

Examples (3.43-3.47)
3.43 ***kumār*** ‘Kumar’
3.44 ***maruttuvar*** ‘Doctor’
3.45 ***ciṅkam*** ‘Lion’
3.46 ***kiḷi*** ‘Parrot’
3.47 ***tēvatay*** ‘Fairy’

3.2.9.2 Inan: inanimate

Nouns that are not alive/personified are inanimate.

Examples (3.48-3.50)
3.48 ***kaṇiṉi*** ‘Laptop’
3.49 ***vīṭu*** ‘House’
3.50 ***maram*** ‘Tree’

3.2.9.3 Hum: human

It is a subtype of animate nouns but includes only human beings and names of professions, not animals or personified animates.

Examples (3.51-3.53)

3.51 ***āciriyar*** ‘Teacher’

3.52 ***cītā*** ‘Sita’

3.53 ***maṉitar*** ‘Human’

3.2.10 Number:number

The number is an inflectional feature, indicating the number of nouns present in the context of the sentence.

3.2.10.1 Sing: singular number

A singular noun indicates ‘one’ entity.

Example (3.54) ***malar*** ‘flower’

3.2.10.2 Plur: plural number

A plural noun indicates several entities.

Example (3.55) ***malarkaḷ*** ‘flowers’

3.2.10.3 Dual: dual number

A dual noun indicates two entities.

Example (3.56) ***iruvar*** 'two persons’

3.2.10.4 Tri: trial number

A trial noun indicates three entities.

Example (3.57) ***mūvar*** ‘three persons’

3.2.10.5 Coll: collective/mass/singulare tantum

Coll is a special type of singular noun which denotes words describing a sets of objects, i.e. semantic plural.

Example (3.58) ***āṭṭu mantay*** 'herd of sheep'

3.2.11 Case: case

Case is an inflectional feature of nouns and derived nouns in Tamil. It helps in identifying the role of nouns or noun phrases in a sentence, especially in free constituent order languages. The following cases are found in Tamil.

3.2.11.1 Nom: nominative/ direct

Nom marks the syntactic subject/ irrational objects of the sentence. The base form of noun without any added case marker.

Examples (3.59 and 3.60)
3.59 ***vilaṅkukaḷ -∅*** *kūṇṭiṉuḷ iruntaṉa* **'Animals** were inside cage'
3.60 ***kumār-∅*** *paḻam cāppiṭṭāṉ* **'Kumar** ate fruits'

3.2.11.2 Acc: accusative/ oblique

Acc marks the object of the sentence in Tamil. *-ay* is the case marker. It is marked for rational objects and optionally marked for irrational objects.

Examples (3.61 and 3.62)
3.61 *nāṉ* ***puttakattayk*** *koṭuttēṉ* 'I gave the **book'**
3.62 *rātā* ***iṭli-∅*** *cāppiṭṭāṉ* 'Radha ate **idli'**

In example (3.62), the object is not case marked. It can still be identified as an object using the transitivity of the verb.

3.2.11.3 Dat: dative

Dat marks the indirect object of the sentence. *-ku* is the case marker. This is possible when ditransitive verbs occur in a sentence. The less affected patient is marked 'iobj'.

Example (3.63)
nāṉ ***appāvukku*** *kaṭitam koṭuttēṉ*
'I gave a letter to my **dad'**

3.2.11.4 Gen: genitive

Genitive case marks the nouns belonging to its governor. That is, it talks about one's possession. *-uṭayya/-iṉ* is the case marker in Tamil.

Examples (3.64 and 3.65)
3.64 *rāmu* ***appāvuṭayya*** *pēṉāvay eṭuttāṉ* 'Ramu took **dad's** pen'
3.65 *nāṉ* ***rātayyiṉ*** *puttakattay vāṅkiṉēṉ* 'I got **Radhai's** book'

3.2.11.5 Voc: vocative

The vocative case is used to address someone and the noun is marked with a long vowel sound at the end of the word in Tamil.

Example (3.66)
ammā! *iṅkē vāruṅkaḷ* '**Mom!**, come here'

3.2.11.6 Ins: instrumental/ instructive

Instrumental case is used for nouns that are used as instruments to do some work. *-āl* is the case marker.

Example (3.67)
cāviyāl *katavayt tiṟantēṉ*
'(I) opened the door **with** the key'

3.2.11.7 Com: comitative/ associative

Nouns referring to comitative or associative case are marked with *-ōṭu/-uṭaṉ* in Tamil. It gives the meaning 'with/along with/together with' in the sentence.

Example (3.68)
*rāmaṉ**uṭaṉ**/rāmaṉ**ōṭu** ceṉṟēṉ* '(I) went **with/along with** Raman'

3.2.11.8 Ben: benefactive/ destinative

The benefactive case expresses the meaning 'for someone/something' in English. In Tamil, it is marked with *-kkāka/kkāṉa*.

Examples (3.69 and 3.70)
3.69 *nāṉ pommaykaḷay makaḷu**kkāka** vāṅkiṉēṉ*
'I got toys **for** my daughter'

3.70 *nāṉ kumāru**kkāṉa** caṭṭayyayk koṭuttēṉ*
'I gave Kumar**'s** shirt to him'

3.2.11.9 Loc: locative

The locative case refers to a location in space or time. *-il* is the locative case marker in Tamil, used for both internal and external location.

Example (3.71)
Location in space
*nāṉ vīṭṭ**il** uḷḷēṉ*
'I am **at** home'

Example (3.72)
Location in time
*nāṉ inta vārat**til** varuvēṉ*
'I will come **(in)** this week'

3.2.11.10 Abl: ablative/ adelative

Ablative/ adelative case denotes the direction 'from' some point, including the source point in Tamil. -*iliruntu* is the case marker used.

Example (3.73)
*nāṉ vaṅkiyi**liruntu** paṇam eṭuttēṉ*
'I took money **from** the bank'

3.2.11.11 All: allative/ adlative

The allative case indicates the direction 'to' something. -*ku* is the case marker seen in Tamil, which is the same as the dative marker. The difference is obtained from the sentential arguments. If the word is an argument of the sentence, it is dative and if it is not, it is Allative.

Example (3.74)
*nāṉ kaṭay**kkup** pōṉēṉ*
'I went **to** the shop'

3.2.12 Definite: definiteness or state

Nouns are either denoted as definite or indefinite based on specificity of the objects.

3.2.12.1 Ind: indefinite

Non-specific nouns are marked as indefinite as particularity is absent. It refers to random noun and not a particular one.

Example (3.75) ***oru*** *maram* **'a** tree'

3.2.12.2 Def: definite

Nouns with specificity refer to a particular object and it is called definite. Determiners help in expressing specificity in Tamil.

Example (3.76) ***anta*** *maram* ‘that tree’

3.2.13 Verbform: form of verb or deverbative

Verbforms in all languages display a variety of forms. Finite, infinite, participle, gerund, converb are the different forms seen in Tamil.

3.2.13.1 Fin: finite verb

Finite verbs are complete in meaning by themselves. If the verb has a non-empty mood, it's considered a finite verb.

Example (3.77) ***vantēṉ*** ‘(I) **came’**

3.2.13.2 Inf: infinitive

Infinite forms are very productive in Tamil, which does not spell out the tense of the action. It is likely denoting futuristic action.

Example (3.78) *malar* ***vara*** *vēṇṭum* ‘Malar should/ needs to **come’**

3.2.13.3 Part: participle, verbal adjective

Participle forms are forms of verbs that are non-finite, sharing the property of both verbs and adjectives. These are also called adjectival participles.

Example (3.79) *nēṟṟu* ***vanta*** *payyaṉ* ‘The boy who **came** yesterday’

3.2.13.4 Ger: gerund

Gerunds in Tamil are considered as nouns in POS as they take up case markers at the word final position like nouns.

Example (3.80)
uṭaṟpayiṟcci ***ceytal/ceyvatu*** *maṉatukku puttuṇarcci tarum*
‘Exercising refreshes the mind’

3.2.13.5 Conv: converb, transgressive, adverbial participle, verbal adverb

Converbs are adverbial participles/ verbal participles that share the properties of verbs and adverbs.

Example (3.81) *rām* ***vantu*** *pōṉāṉ* 'Ram **came** and went'

3.2.14 Mood: mood

Mood of the verb expresses modality and it is expressed by auxiliaries in Tamil. It is sub-classified as indicative, imperative, conditional, potential, desiderative, and necessity in Tamil.

3.2.14.1 Ind: indicative

Indicative is considered as the default mood. Statements and everyday activities are stated in an indicative mood without adding any attitude of the speaker.

Example (3.82) *avaḷ* ***ceytāḷ*** 'she **did** (it)**'**

3.2.14.2 Imp: imperative

The speaker orders someone to do an action. It is a commanding mood and not a request.

Example (3.83) *nī* ***cey*** 'you **do** (it)**'**

3.2.14.3 Cnd: conditional

Conditional mood expresses the situations where the action must have happened in certain circumstances but it didn't happen. In Tamil, circumstantial actions are expressed using conditional verbs.

Example (3.84) *nī* ***vantāl*** *nāṉ varuvēṉ* 'I will come if you **come'**

3.2.14.4 Pot: potential

Certainty of action is not seen in potential mood. In English, modal verbs like ‘may, can’ express this mood. But in Tamil, ***-āṁ*** marker at the word final position expresses the same.

Example (3.85) *kiraṉ vīṭṭiṟku pōkl**āṁ*** ‘Kiran **can** go home’

3.2.14.5 Des: desiderative

The auxiliary verb *vēṇṭum* ‘wants to’ expresses the mood of desideration.

Example (3.86) *eṉakku vīṭṭiṟku pōka **vēṇṭum*** ‘I **want to** go home’

3.2.14.6 Nec: necessitative

English modal verbs must/should/have to express necessity, for which Tamil equivalent is *vēṇṭum*

Example (3.87) *eṉakku vīṭṭuppāṭattay muṭittāka vēṇṭum*
‘I have to complete myhomework’

3.2.15 Tense: tense

Tense is a main feature of the POS verb. It can also be present in POS tagged nouns/ adjectives in Tamil. Present, past and future are the subclasses of tense seen in Tamil.

3.2.15.1 Past: past tense/ preterite/ aorist

The actions that happened before a referral time point are classified under past tense. *-t/-ṉ/-nt* are the common markers found in Tamil past tense verbs.

Example (3.88) *kumār **vantāṉ*** ‘Kumar **came’**

3.2.15.2 Pres: present/ non-past tense/ aorist

The actions that are progressing at the time of speech are classified as present tense. *-kiṟu, kiṉṟu* are commonly seen Tamil present tense markers.

Example (3.89) *kumār* ***varukiṉṟāṉ/ varukiṟāṉ*** 'Kumar **is coming'**

3.2.15.3 Fut: future tense

The actions that happen after a referral time point are called future tense. *-p/-v* are the common future tense markers seen in Tamil.

Example (3.90) *kumār* ***varuvāṉ*** **'**Kumar **will come**'

3.2.16 Aspect: aspect

Aspect is a feature of verbs, which is extended to POS nouns and adjectives as well in Tamil. It denotes the duration of action/ verb in time. Progressive, perfective and prospective are the aspects seen in Tamil.

3.2.16.1 Perf: perfect aspect

The action is completed in the recent/remote past, referring to a specific point of time of completed action. *-iru* is a perfective marker in Tamil verbs.

Example (3.91) *malar vant****irukkiṟāḷ*** 'Malar **has** come'

3.2.16.2 Prog: progressive aspect

The action that is progressing or happening at the time of speech is denoted by progressive aspect. *-koṇṭiru* is the progressive marker inTamil, used for past, present and future tense.

Example (3.92) *malar vantu****koṇṭiru****kkiṟāḷ* 'Malar **is** com**ing**'

3.2.17 Voice:voice

Voice is a feature of the verb, extended to nouns and adjectives in Tamil. Voice in Tamil can be active/passive or causative.

3.2.17.1 Act: active or actor-focus voice

The sentential subject is the agent or performer of the action. The object of the sentence is affected by the action, which becomes the patient.

Example (3.93) *kumār vīṭṭay* ***cuttam ceytāṉ*** *'*Kumar **cleaned** the house'

3.2.17.2 Pass: passive or patient-focus voice

The sentential subject is affected by the action and becomes the patient. The performer of the action may or may not be present in the sentence. *-paṭu* is the passive marker in Tamil verbs.

Example (3.94) *vīṭu cuttam* ***ceyyappaṭṭatu*** 'The house **was cleaned**'
Here, the agent of the verb is absent.

Example (3.95) *vīṭu kumārāl cuttam* ***ceyyappaṭṭatu*** 'The house was cleaned by Kumar'
The doer of the action or oblique agent is present here with the instrumental case marker *-āl.*

3.2.17.3 Cau: causative voice

Causative voice semantics vary from active or passive voice as the number of people involved in the action are more in number when it comes to causative constructions.

Example (3.96) *nāṉ kumāray vīṭu cuttam ceyya****vayttēṉ*** 'I made Kumar clean the house'

3.2.18 Polarity: polarity

Polarity is the feature of verbs in Tamil sentences, which denotes whether the sentence is positive or negative.

3.2.18.1 Pos:positive, affirmative

The presence of an action or object is denoted by positive polarity.

Example (3.97) *nāṉ* ***vantēṉ*** 'I **came'**

3.2.18.2 Neg: negative

The absence of an action or object is denoted by negative polarity.

Example (3.98) *nāṉ varav**illay*** 'I **did not** come'

3.2.19 Person: person

Person is morphologically seen in Tamil verbs, which has agreement features with the subject of the sentence. In subjectless constructions, the 'person' information encoded in the verb still covers the meaning of pronouns in its absence.

3.2.19.1 1: first person

First person refers to the speaker of the sentence. In plural constructions, the speaker and one or more other persons are included in the first person.

Example (3.99) *(nāṉ) vant**ēṉ*** '(I) came'

3.2.19.2 2: second person

Second person refers to the addressee of the narration or the text. Pluralisation of second person subject' object or oblique form of nouns is possible.

Example (3.100) (*nī*) *vant**āy*** '(you) came'

3.2.19.3 3: third person

Third person refers to the other persons apart from the addressee and the speaker of the text or narration.

Example (3.101) (*avaṉ) vant**ā̲ṉ*** ‘(he) came’

3.2.20 Polite: politeness

Politeness is used in Tamil to mark the feature of respect towards a person or some people.

3.2.20.1 Form: formal register

Formal register can be seen for both singular and plural nouns/ pronouns. Singular noun/pronoun takes up honorific marker to mark respect. Plural nouns or pronouns take up the same honorific marker to either show plurality/ respect or both.

Example (3.102) ***nīṅkaḷ*** *vantīrkaḷ* ‘You (Sg) came’
It is marked with respect in the above example.

3.2.21 Clusivity

Clusivity is a feature of pronouns in Tamil. It can occur in both subject and object positions.

3.2.21.1 In: inclusive

Inclusive feature includes the second person or the listener of the text/ narration.

Example (3.103) ***nām*** ‘we’ (I+you)

3.2.21.2 Ex: exclusive

Exclusive feature excludes the second person or the listener of the text/ narration.

Example (3.104) ***nāṅkaḷ*** ‘we’ (I+they)

3.3 Parts Of Speech guidelines

All linguistic frameworks believe that words can be categorized based on word class or Parts Of Speech (POS) with respect to individual language’s behaviour. In the thesis, the POS guidelines have been designed using Universal Dependency (UD) tags. This section describes the different

types of tags that are needed to tag the Tamil data set. The rules developed for each tag are language specific. This set of POS guidelines was framed based on the syntactic and semantic structure of the sentence. It has described a set of 16 tags with examples as seen below:

Open class tags: ADJ, ADV INTJ, NOUN, PROPN and VERB
Closed class tags: ADP, AUX, CCONJ, DET, NUM, PART, PRON and SCONJ
Other: PUNCT, SYM

3.3.1 Open class tags

Open class tags are lexical/ content words which are very productive and new words are added to the list frequently.

3.3.1.1 ADJ: adjective

Adjectives are words that add information to nouns or modify a noun by specifying its property. All the word finals with *-āṉa* are tagged as adjectives as they describe the quality of a noun. Also, words describing the state of being like colour, shape, size, time, etc. are classified as ADJs.

(i) Attributive adjective

Attributive adjectives in Tamil occur preceding the noun and it is not separated from the noun by the linking verb.

(a) Quality

It describes the basic character or nature of a noun.

Example (3.105)

3.105	*ūṭṭi*	*oru*	***aḻakāṉa***	*malayppakuti*
	ooty-NOM	one	**beautiful**	hill station-NOM

'Ooty is a **beautiful** hill station'

Example (3.106)

3.106 *mutalaykaḷ* ***kūrmayyāṉa*** *paṟkaḷ* *ko(ḷ)-ṇṭ-avay*

crocodiles-NOM **sharp** teeth has-PST-3.PL.N

'Crocodiles have **sharp** teeth'

(b) size/quantity

It measures the noun in terms of its dimensions and amount.

Example (3.107)

3.107 *kaṭaṟkarai-yil* ***ciṟiya*** *caṅkukaḷ* *uḷḷa-ṉa*

beach-LOC **small** shells there-3.PL.N

'There are small shells at the beach'

Example (3.108)

3.108 *aṅku* ***niṟayya*** *iṉippu* *vakaykaḷ* *uḷḷa-tu*

there **many** sweet varieties present-3.SG.N

'Many sweet varieties are present there'

(c) Shape/ colours

It describes the shades/tones and the forms in which nouns occur.

Example (3.109)

3.109 *rāmu* ***civappu*** *rōjā.v-ay* *vāṅk-iṉ-āṉ*

ramu-NOM **red** rose-ACC buy-PST-3.SG.M

'Ramu bought red roses'

Example (3.110)

3.110 *paḻaṅkaḷ* ***uruṇṭay,*** ***nīḷvaṭṭam,*** ***vaṭṭa*** *vaṭiva.ṅ-kaḷ-il* *kāṇa.p-paṭ-um*

fruits **sphere,** **oval,** **circle** shape-PL-LOC find-PASS-3.SG.N

'Fruits are found in spherical, oval and circle shapes.'

(d) Age/time

The age related information of a noun is indicated by adjectives in Tamil.

Example (3.111)

3.111 *rāji* *oru* ***putu*** *pēṉā.v-ayk* *koṇṭuva-nt-āḷ*

raji-NOM one **new** pen-ACC bring-PST-3.SG.F

'Raji bought a new pen'

(e) Emotions

In Tamil, human emotions like happiness, sadness, anger, excitement are expressed through adjectives as well.

Example (3.112)

3.112 *itu* *oru* ***makiḻcci.y-āṉa*** *ceyti*

it-NOM one **happy-ADJ** news-NOM

'It is a happy news'

(f) Adjectives as intensifiers

The tag ADJ is also used for words which intensify nouns, when it precedes the noun.

Example (3.113)

3.113 *vikṉēṣ* *vēlay* *cey.v-at-il* ***mika.c*** *ciṟanta.v-aṉ*

vignesh-NOM work-NOM do-FUT.AP-3.SG.N-LOC **very** good-3.SG.M

'Vignesh is very good at doing work'

(ii) Adjectival modifiers of adjectives

The tag ADV is used to describe modifiers of adjectives in general. But here, the tag ADJ is used for modifiers of adjectives in specific occurrences as seen below:

(a) ordinal numeral modifiers of an adjective

The cardinal numbers like onṟu 'one', iraṇṭu 'two' tagged as NUM. But when it comes to ordinal numbers, especially occurring in the adjectival position, it is given the tag, ADJ. These define the position of the noun.

Example (3.114)

3.114 *celvi* ***iraṇṭ-āvatu*** *aṟay.y-il* *uḷḷ-āḷ*

Selvi-NOM two-ORD room-LOC be-3.SG.F

'Selvi is in second room'

(b) Occurrence of pair of adjectives

When a pair of adjectives consecutively, the first modifier of the noun (adjective) is still tagged as ADJ.

Example (3.115)

3.115	*cītā*	***maṟṟa***	*nalla*	*kāy-kaḷ-ay.p*	*pār-tt-āḷ*
	Sita-NOM	other	good	vegetable-PL-ACC	see-PST-3.SG.F

'Sita saw other good vegetables'

Syntactic cue

(i) All *-āṉa* ending words are tagged as ADJ

(ii) The adjectives are either followed by a NOUN/PRON or another ADJ.

Test case

-avaṉ, *-avaḷ* and *-atu* can be added to the word to check if a grammatical word is formed. If it is formed, it is an adjective.

(Note: All predicative adjectives and *-kkāṉa* ending words are tagged as NOUN)

3.3.1.2 ADV: adverb

Adverbs are modifiers of verbs. It adds information on time (non-nominals), place or manner of the action performed. All the words ending with -āka are tagged as ADV. Adverbs also modify adjectives and other adverbs. All *-kkāka* ending words are tagged NOUNs and not ADVs.

Example (3.116)

3.116	*kītā*	***vēkam-āka***	*ōṭ-iṉ-āḷ*
	geetha-NOM	**fast-ADV**	run-PST-3.SG.F

'Geetha ran fast'

Adverbs are subdivided into different categories:

(i) Interrogative/relative adverbs

This type includes all the question words in Tamil. This tag includes circumstantial usage as well, which is neither interrogative or relative.

Example (3.117) and (3.118)

3.117 *avarkaḷ* ***eṅkē*** *pō-n̲-ārkaḷ*

they-NOM **where** go-PST-3.PL.M/F(Hon)

'Where did they do?'

3.118 *nī* ***evvāṟu*** *va-nt-āy* *eṉa.t* *teri.y-a-v.illay*

you-NOM **how** come-PST-2.SG.M/F COMP know-INF-NEG

'(I) don't know how you have come'

List of interrogative adverbs: *eṅkē, eppoḻutu, eppaṭi, ēn̲, evvāṟu, etaṟku*

(ii) Demonstrative adverbs

Demonstrative adverbs describe time (non-nominals), place, manner and degree. Each category is described below:

(a) Adverb of time (non-nominals)

Adverb of time does not include the words which act like a noun. For instance, *nēṟṟu* 'yesterday', *iṉṟu* 'today', *nāḷay* 'tomorrow' are tagged as NOUN. It includes words like *ippoḻutu* 'now', *appoḻutu* 'then', ēṟkaṉavē 'already', etc.

Example (3.119)

3.119 *kumār* ***ippoḻutu*** *tāṉ* *va-nt-āṉ*

kumar-NOM **now** only come-PST-3.SG.M

'Kumar came now only'

(b) Adverb of place

Adverb of place includes words like *iṅkē* 'here' and *aṅkē* 'there'.

Example (3.120)

3.120 *nāṉ* ***aṅkē*** *pō-ki-ṟēṉ*

I-NOM **there** go-PRES-1.SG.M/F

'I am going there'

(c) Adverb of degree

Adverb of degree describes the strength of an adjective or the degree of work done.

Example (3.121)

3.121 *kumār* ***mika.v-um*** *kaṭiṉam-āka* *uḻay-kkiṟ-āṉ*

kumar-NOM **very** hard-ADV work-PRES-3.SG.M

'Kumar is working very hard'

(d) Adverb of manner

Adverb of manner expresses the behaviour of the action or how the action is performed.

Example (3.122)

3.122 *avarkaḷ* ***metu.v-āka*** *naṭa-ntu* *va-nt-aṉar*

they-NOM **slow-ADV** walk-CPM come-PST-3.PL.N(Hon)

'He is working very hard'

(iii) Indefinite adverbs

The indefinite adverbs do not provide exact time and place information. Uncertainty is seen in this category. It includes, *eṅkēyō* 'anywhere', *eppoḻutō* 'anytime', *eṅkēyāvatu* 'anywhere', *eppoḻutuvēṇṭumāṉālum* 'anytime'.

Example (3.123)

3.123 *nāṅkaḷ* ***eppōtō*** *canti-tt-ōm*

we-NOM **anytime** meet-PST-1.PL.M/F

'We met sometime back'

(iv) Adverbs of frequency

Every action occurs in a certain interval of time. Such a time gap is expressed by adverbs of frequency. It can be definite or indefinite.

(a) Totality adverbs/ adverbs of indefinite frequency

The totality adverbs include words like *eṅkēyum* 'anywhere and…', *eppoḻutum* 'anytime and…' which are not definite.

Example (3.124)

3.124 *kumār* ***eppōtum*** *tūṅk-i.k-koṇṭē* *iru-nt-āṉ*

kumar-NOM **always** sleep-CONT be-PST-3.SG.M

'Kumar was always sleeping'

(b) Adverbs of definite frequency

Adverbs of definite frequency define the time precisely and also express the action in a definite manner.

Example (3.125)

3.125	*nāṉ*	***tiṉamum***	*kālay.y-il*	*uṭaṟpayiṟci*	*cey-v-ēṉ*
	I-NOM	**daily**	morning-LOC	exercise-NOM	do-FUT-1.SG.M/F

'I exercise in the morning everyday'

(v) Conjunctive adverbs

The conjunctive adverbs are also called adverbs of purpose or adverbs of reason as they state the reason/purpose of action and perform the function of connecting clauses as well.

Example (3.126)

3.126	*kumār*	*va.r-a.v-illay* **,**	***ataṉāl,***	*nāṉ*	*cīkkiram*	*pō-ṉ-ēṉ*
	kumar-NOM	come-INF-NEG	**so**	I-NOM	soon	go-PST-1.SG.M/F

'Kumar didn't come, so I went early.'

Note: If the same word *ataṉāl* 'so' occurs at the beginning of a sentence, it is tagged as SCONJ.

(vi) Focusing adverbs

Focusing adverbs emphasize a particular action or specific part of a clause / sentence.

Example (3.127)

3.127	*avarkaḷ*	***maṭṭum***	*muṭi.kk-a-vēṇṭum*
	they-NOM	**only**	complete-INF-should

‘Only they should complete’

(vii) Negative adverbs

Negative adverbs indicate that the action has not happened or indicate that it has not happened anywhere or anytime

Example (3.128)

3.128	*nāṉ*	***orupōtum***	*tavaṟu*	*cey-t-at-illay*
	I-NOM	**never**	wrong	do-PST.AP-3.SG.N-NEG

‘I never did something wrong’

Syntactic cue

(i) words ending with *-āka*

(ii) It is followed by a verb or an adjective or another adverb.

Test case

The sentential order of adverbs can be changed, retaining the grammaticality and the meaning of the sentence/clause.

3.3.1.3 INTJ: interjection

Interjections are words that express exclamation or emotional reaction in the form of sounds that are not a part of the language’s dictionary. These words are found more in stories or real life conversations, when compared to scientific data.

In case, if the expressing word originally belongs to some other category, then it remains the same. Only expressive words with no category, fall in INTJ.

Example (3.129) and (3.130)

3.129 ***āhā!*** *eṉṉa* *cuvay.y-āṉa* *kāppi*

Aww! what taste-ADJ coffee-NOM

'Aww! What a tasty coffee!'

3.130 ***ayyō!*** *inta* *ceyti* *kaṭum* *vētaṉay* *aḷi-kkiṟ-atu*

alas! this news-NOM severe agony give-PRES-3.SG.N

'Alas! This news gives severe agony'

3.3.1.4 NOUN: noun

Nouns are a part of speech typically denoting a common thing, animal, plant or idea. In Tamil, gerunds share the quality of noun and it is tagged as NOUN.

Example (3.131)

3.131 ***paḻa.ṅ-kaḷ-il*** *niṟayya* ***vakay-kaḷ*** *uḷḷa-ṉa*

fruit-PL-LOC many **variety-PL** be-3.PL.N

'There are many varieties in fruits'

Some of the following special cases are considered as NOUN:

(i)Gerunds

All gerunds with or without case marker is considered as NOUN

Example (3.132)

3.132 *mūccuppayi̲rci* ***ceytal*** *uṭalnala.tt-i̲rku* *nalla-tu*

breathing exercise **doing** health-DAT good-3.SG.N

'Doing breathing exercises is good for health'

(ii) Nominalised participle verbs

Nominalised participle verbs like *iruntava̲n* 'he who was there', *irukki̲rava̲n* 'he who is there', *iruppava̲n* 'he who is there (regularly)', *irukkātava̲n* 'he who is not there' and followed by any case marker are clubbed under one category.

Example (3.133)

3.133 *nē̲r̲ru* ***va-nta-va̲n-ay*** *nā̲n* *pār.kk-a.v-illay*

yesterday **come-PST.AP-3.SG.M-ACC** I-NOM see-INF-NEG

'I didn't see who had come yesterday'

(iii) Oblique forms

In Tamil, the oblique forms of words are considered as nouns rather than adjectives.

Example (3.134)

3.134 ***pu̲nita*** *nīr*

sacred water

'Sacred water'

Here, the root word of *pu̲nita* is *pu̲nitam* 'sacred'. So, *pu̲nita* is in oblique form. Such cases are considered as nouns and not adjectives.

(iv) Nouns of Space and Time (NST)

All directions are included in NOUNs as they belong to Nouns of space and time. Also, Nouns of time like *iṉṟu* 'today', *nēṟṟu* 'yesterday', *nāḷay* 'tomorrow', *aṉṟu* 'then' are considered as nouns.

Example (3.135)

3.135	***kiḻakku***	*tamiḻaka.tt-il*	***iṉṟu***	*maḻay*	*pey-y-um*
	east-NOM	Tamilnadu-LOC	**today-NOM**	rain-NOM	fall-FUT-3.SG.N

'Eastern Tamil Nadu will have rainfall today'

(v) Predicative adjective

The adjective acts like a noun at the end of a sentence in Tamil.

Example (3.136) and (3.137)

3.136	*anta*	*maruttuvar*	*mikavum*	***nalla-var***
	that	doctor-NOM	very	good-3.SG.M(Hon)

'That doctor is very good'

3.137	*anta*	*payyaṉ*	***aḻak.āṉa-vaṉ***
	that	boy-NOM	handsome.NOM-ADJL-3.SG.M

'That boy is handsome'

Note: adverbs end with *-āka* but nouns end with *-kkāka*

Example: *amaytiyāka* 'quietly' is ADV and *amaytikkāka* 'for peace' is NOUN

3.3.1.5 PROPN: proper noun

Proper nouns refer to a specific individual, place, or an object. It is a subclass of NOUN and retains the syntactic properties of nouns.

Example (3.138)

3.138	***koṯaykkāṉal***	*oru*	*nalla*	*cuṟṟulāt*	*talam*	*āk-um*
	Kodaikanal-NOM	a	good	tourist	place-NOM	be-3.SG.N
	'Kodaikanal is a good tourist place'					

Some special cases of PROPN include:

(i) Multi word names

Multi word names like ***aykkiya arapu*** *nāṭukaḷ* 'United Arab Emirates' are tagged as PROPN for the specific word and NOUN for common word. Here, *aykkiya arapu* is tagged as PROPN and *nāṭukaḷ* is tagged as NOUN

Example (3.139)

3.139	*nāṉ*	***meriṉā***	*kaṭaṟkaray-kku.c*	*ce-ṉṟ-ēṉ*
	I-NOM	**Marina-NOM**	beach-DAT	go-PST-1SG.M/F
	'I went to Marina beach'			

Here, *meriṉā* is tagged PROPN and *kaṭaṟkaraykkuc* is marked NOUN.

(ii) Acronyms

Acronyms of proper nouns like ***aynā*** 'UN', ***yuṉeskō*** 'UNESCO' are tagged as PROPN.

Example (3.140)

3.140 *nāṉ* ***yuṉicep*** *mānāṭṭ-il* *paṅkēṟ-kiṟ-ēṉ*

I-NOM **UNICEF-NOM** conference-LOC participate-PRES-1SG.M/F

'I am participating in the UNICEF conference'

(iii) Symbols:

Letters with symbols specifying a product name are considered as PROPN.

Example (3.141)

3.141 ***tanā-81*** *eṉpatu* *tirucci* *māvaṭṭatt-ay.k* *kuṟi-kkiṟ-atu*

TN-81 means Trichy-NOM district-ACC mark-PRES-3SG.N

'TN-81 refers to Trichy district'

3.3.1.6 VERB: verb

Verb is a lexical word that expresses actions done by the subject. Verbs become the syntactic root of the sentence, governing the number and type of constituents in a clause. Morphologically, it encodes the tense, aspect, mood, person, number, gender and voice information in Tamil. The encoded information is expressed inflectionally by the VERB. Sometimes, particles (PART) or auxiliaries (AUX) are also used to express the same.

Example (3.142)

3.142 *nāṉ* *eṉ* *vīṭṭuppāṭa.tt-ay.c* ***cey-t-ēṉ***

I-NOM my homework-ACC **do-PST-1SG.M/F**

'I did my homework'

Note: modal verbs are categorized as AUX and gerunds are categorized as NOUN in Tamil.

Different forms of verbs are observed in Tamil as follows:

(i) Participles

Participles are a form of verbs that share the property of adjectives and verbs. The sentence is not complete with participles.

Example (3.143)

3.143	*nāṉ*	*vīṭṭuppāṭa.tt-ay.*	***eḻut-i***	*muṭi-tt-ēṉ*
	I-NOM	homework-ACC	**write-CPM**	complete-PST-1.SG.M/F

'I completed writing my homework'

(ii) Infinitives

Infinitive constructions are special verbs which are very productive in Tamil. It is considered as VERB in Tamil. The verb always ends with *-a* and it is always followed by another verb or auxiliary to complete the meaning of the sentence.

Example (3.144)

3.144	*eṉ.ṉ-āl*	*nilāv-ay.p*	***pār.kk-a***	*muṭi-kiṟ-atu*
	I-INS	moon-ACC	**see-INF**	can-PRES-3.SG.N

'I can see the moon'

3.3.2 Closed class tags

Closed class words are functional words and thus, they are mostly fixed in number

3.3.2.1 ADP: adposition

Adposition is the term used for both prepositions and postpositions. Tamil is a postpositional language and ADP is used as a tag for all the occurrences. It usually occurs after a Noun Phrase (NP)/ NOUN/ PRON. It results in a single structure expressing the grammatical and semantic relationship within the clause. Most of the adpositions in Tamil are found to be grammaticalized from verbs like *iruntu* 'from', *varay* 'till', etc.

Example (3.145) and (3.146)

3.145	*mara.tt-il*	***irunthu***	*paḻam*	*viḻu-nt-atu*
	tree-LOC	**from**	fruit-NOM	fall-PST-3.SG.N
	'The fruit fell from the tree'			

3.146	*pārvayyāḷar-kaḷ*	*8.00*	*maṇi*	***mutal***	*6.00*	*maṇi*	***varay***	*va.r-alām*
	visitor-PL	8.00	o'clock	**from**	6.00	o'clock	**till**	come-HORT
	'Visitors can come from 8 am to 6 pm '							

List of ADP's

aṭiyil, appāl, arukilēyē, arukē, ākac, iṭayyil, iṭayyē, iṉṟi, iruntu, iruntē, uṭpaṭa, uṭpaṭṭa, uḷ, uḷḷa, uḷḷē, etiretirē, etirē, ēṟpa, oṭṭi, kīḻē, kīḻ, kīḻk, kuṟittu, kuṟukkē, kūṭavē, koṇṭa, koṇṭu, cuṟṟi, cuṟṟiyum, cuṟṟiyuḷḷa, tavira, naṭuvil, naṭuvē, paṟṟi, paṟṟiya, paṟṟiyum, piṉṉāl, piṟaku, pōtu, pōtum, mattiyil, mītu, mutal, muṉ, muṉṉāl, muṉpu, mūlamō, mūlam, mēlē, mēl, varay, varayyilum, viṭa, etc.

Syntactic cue

It is preceded by NP/noun/pronoun (defining the position of the noun)

3.3.2.2 AUX: auxiliary

Auxiliaries are functional words that add tense, aspect, mood, person, number, gender and voice information to the main verb. Sometimes, in Tamil, auxiliaries act as lexical or main verbs. Auxiliaries include copulas and modal verbs as well, even though they are not very productive in Tamil. The modality information is mostly expressed by the lexical verb and a few words are found which are marked as AUX.

Example (3.147)

3.147	*nāṉ*	*va.r-a*	***vēṇṭum***
	I-NOM	come-INF	**should**
	'I should come'		

Different kinds of AUX are seen below:

(i) Modal auxiliaries

Some modal verbs are counted as auxiliaries in Tamil.

Example (3.148)

3.148	*hari*	*taṉ*	*vēlay.y-ay.p*	*pār.kk-a*	***vēṇṭum***
	hari-NOM	his	work-ACC	see-INF	**should**
	'Hari should do his work'				

(ii) Tense auxiliaires

Tense auxiliaries express when the action is taking place. It is especially seen while expressing continuous tense in Tamil.

Example (3.149)

3.149	*hari*	*vēlay.y-ay*	*cey-tu*	***koṇṭu***	*iru-kki̲r-ā̲n̲*
	hari-NOM	work-ACC	do-CPM	**CONT**	be-PRES-3.SG.M

'Hari is doing the work'

Note: In some cases, *koṇṭu* is the main verb (VERB) and not AUX as seen below:

Example (3.150)

3.150	*ravi*	*pāl-ay.k*	***koṇṭu***	*va-nt-ā̲n̲*
	ravi-NOM	milk-ACC	**bring**	come-PST-3.SG.M

'Ravi brought the milk'

(iii) Passive auxiliaries

The passive auxiliaries are identified by *-paṭu* marker in Tamil. It should be split from the lexical verb if the auxiliaries are found along with it.

Example (3.151)

3.151	*ulakam-ē*	*korō̲n̲ā*	*to̲r̲r̲-āl*	*pāti.kk-a.p-**paṭṭu***	*iru-kki̲r-atu*
	world-EMPH	corona-NOM	disease-INS	affect-INF-**PASS**	be-PRES-3.SG.N

'The whole world is affected by Corona disease'

(iv)Verbal copulas

In Tamil, not many copulas are found. A few of those copulas are listed under auxiliaries.

Example (3.152)

3.152	*ravi*	*oru*	*māṇavaṉ*	***ā-v-āṉ***
	ravi-NOM	a	student-NOM	**become.COP-FUT-3.SG.M**
	'Ravi is a student'			

List of auxiliaries

iru, vēṇṭu, koḷ, viṭu, māṭṭu, vā, vay, kūṭu, paṭu, muṭi, cel, cey, uḷḷa, ōṭu, iḻu, illay, alla, pār, etc.

Note: In cases like *uṭpaṭṭatu* 'subjected to'*, veḷippaṭuttum* 'reveal'*,* etc., *paṭu* is not a passive marker. It's the lexical word as a whole.

3.3.2.3 CCONJ: coordinating conjunction

Coordinating conjunctions are lexical or clausal connectors without syntactic subordination. These words exhibit the semantic relationship between the two or more clauses.

Example (3.153) and (3.154)

3.153	*nāṉ*	*pū*	***mārrum***	*paḻam*	*vāṅk-i-va-nt-ēṉ*
	I-NOM	flower-NOM	**and**	fruit-NOM	buy-CPM-come-PST-1.SG.M/F
	'I bought fruit and flower'				

3.154	*eṉ.a-kku*	*āppiḷ*	***allatu***	*ārañcu*	*vēṇṭum*
	I-DAT	apple-NOM	**or**	orage-NOM	want
	'I want apple or orange'				

List of CCONJs

mā̲r̲rum, allatu, ā̲ṉāl , etc.

3.3.2.4 DET: determiner

Determiners are functional words which convey the reference point of the noun/ noun phrase to the context. It indicates whether the noun is referring to a definite or indefinite element or a closer or farther element, or if the word is referring to the whole entity, etc.

The following types are found in Tamil:

(i) Demonstrative determiners

Demonstrative determiners occur right before the noun, either demonstrating the noun or introducing the following noun in a sentence. It includes numbers like *anta* 'that'*, inta* 'this'*, avay* 'those'*, ivay* 'these', etc.

Example (3.155)

3.155	***anta***	*payya̲ṉ*	*va.r-uv-ā̲ṉ*
	that	boy-NOM	come-FUT-3.SG.M
	'That boy will come'		

(ii) Interrogative determiners

Interrogative determiners modify a noun in the form of direct or indirect questions.

Example (3.156)

3.156	*nīṅ-kaḷ*	***enta***	*nāṭṭi̲r̲-ku.c*	*cel-ki̲r̲-īrkaḷ*
	you-PL	**which**	country-DAT	go-PRES-2.PL.M/F
	'Which country are you going to?'			

(iii) Quantity determiners

Quantity of the noun is determined by the quantity determiners.

Example (3.157)

3.157	*aṅku*	***cila***	*paṭṭāmpūcci-kaḷ*	*uḷḷ-aṉa*
	there	**a few**	butterfly-PL.NOM	be-3.PL.N

'There are a few butterflies'

List of DETs

ak, anta, aṉayttu, aṉayvarukkum, ap ,ik, itu, it, inta, ip, im, iru, enta, ellām, oru, ovvoru, cila, pala, palavaṟṟiṉ, palvēṟu, piṟa, muṟṟilum, muḻu, vēṟu, etc.

3.3.2.5 NUM: numeral

A numeral is a functional word, expressing a number in the form of quantity, sequence, frequency or fraction. All the cardinal numbers in the form of numbers or words are included under NUM. It includes date/time, phone numbers, counting numbers, etc.

Alphanumeric characters are not included under NUM.

Example (3.158)

3.158	*nāṉ*	***3***	*maṇi-kku*	*va-nt-ēṉ*
	I-NOM	**3**	time-DAT	come-PST-1.SG.M/F

'I came at 3 o'clock'

List of NUMs

- Numbers/digits: 0, 1, 2, 3, 4, 5, 2014, 1000000, 3.14159265359
- Date/time: 11/11/1918, 11:00
- Word forms: *oṉṟu, iraṇṭu, mūṉṟu,eḻuppatu ēḻu*

- Tamil numerals: *ka* (1)
- Roman numerals: I, II, III, IV, V, MMXIV

3.3.2.6 PART: particle

Particles are functional words which are linked to the preceding word (in Tamil) to complete the meaning of the sentence. Particles are not inflected in Tamil. In general, the PART tag should be used restrictively and only when no other tag is possible. In Tamil, it is generally tagged after a participle form of a verb; -um marker to express clusivity after an open class category of words.

Example (3.159) and (3.160)

3.159	*nāṉ*	*var-um*	***pōtu***	*avaṉ*	*tūṅki-viṭṭ-āṉ*
	I-NOM	come-CPM	**that time**	he-NOM	sleep.CPM-PERF-3.SG.M
	‘He had slept when I came’				

3.160	*hari*	*iḷamayy-āka.v-**um***	*aḻak-āka.v-**um***	*iru-kkiṟ-āṉ*
	hari-NOM	young-ADV-**INCL**	handsome-ADV-**INCL**	be-PRES-3.SG.M
	‘Hari is young and handsome’			

List of PARTs

ām, il, um, kaḷiṉ, kaḷil, kūṭiya, kkum, tāṉ, piṟaku, pōtu, etc.

3.3.2.7 PRON: pronoun

Pronouns are words that replace NOUNs or NPs, whose meaning is understood from the textual context.

(i) In Tamil, non-possessive personal, reflexive or reciprocal pronouns are always tagged PRON.

(ii) Possessive pronouns have varied occurrences across languages. In Tamil, they are more like a normal personal pronoun in genitive (with or without case marker -*iṉ*), or a personal pronoun with an adposition -*uṭaṉ*; they are tagged PRON.

(i) personal pronouns

Personal pronouns behave like a noun replacing the proper name of a person.

Example (3.161)

3.161	***avarkaḷ***	*nāṇayaṅ-kaḷ-ay*	*acciṭṭ-aṉ-ar*
	they-NOM	coin-PL-ACC	print-PST-3.PL.M/F
	‘They printed the coins’		

List: *nāṉ, nām, nī, nīṅkaḷ, avaṉ, avaḷ, avar, avarkaḷ,* etc.

(ii) reflexive pronouns

Reflexive pronouns point back to the same noun/pronoun mentioned earlier.

Example (3.162)

3.162	*kamalā*	***taṉakkuttāṉ-ē***	*ciri-ttu-kkoṇṭ-āḷ*
	kamala-NOM	**herself-EMPH**	laugh-CPM-REFL-3.SG.F
	‘Kamala laughed to herself’		

List: *taṉakkuttāṉē, eṉṉayyē, avaḷayyē, avaṉayyē,* etc.

Syntactic cue:

-koḷ is found verb when reflexive pronoun occurs.

(iii) interrogative pronouns

Pronouns that are used to raise questions are classified under interrogative pronouns.

Example (3.163)

3.163 *nī* ***yār***

you-NOM **who**

'Who are you?'

List: *yār, yāruṭayya, eṉṉa*

(iv) possessive pronouns

Possessive pronouns express one's possession/ belonging.

Example (3.164)

3.164 *pū* ***eṉ.ṉ-uṭayya*** *pay.y-il* *iru-nt-atu*

flower-NOM **I-POSS** bag-LOC be-PST-3.SG.N

'It was in my bag'

List: *eṉṉuṭayya, nammuṭayya, avarkaḷuṭayya, uṉṉuṭayya,* etc.

(v) attributive possessive pronouns

Attributive possessive pronouns are independent possessive pronouns which are also called possessive adjectives.

Example (3.165)

3.165 ***uṉ*** *kaṇ-kaḷ-ay* *mūṭu*

your-NOM eye-PL-ACC close.IMP

'Close your eyes'

List: *eṉ/eṉatu, uṉ/uṉatu, nam/namatu, avaratu, avaḷatu, avaṉatu, ataṉatu*, etc.

3.3.2.8 SCONJ: subordinating conjunction

Conjunctions that make clausal constructions where one clause falls as subordinate to the other are SCONJs. It marks the subordinate clause of the sentence.

(i) Complementizers

Complementizers are words that make a whole clause as the subject/object of the sentence.

Example (3.166)

3.166	*hari*	*va.r-u.v-āṉ*	***eṉru***	*eṉ.a-kku.t*	*teri.y-um*
	hari-NOM	come-FUT-3.SG.M	**COMP**	I-DAT	know-3.SG.N.(default)

'Hari had slept when I came'

(ii) Simultaneous construction

When two actions happen at the same time, SCONJs act as the connecting constituent of the two actions (expressed as two clauses).

Example (3.167)

3.167	*kaṇēṣ*	*paṭi-ttuk-koṇṭiru-nta*	***poḻutu/pōtu***	*nāṉ*	*camay-tt-ēṉ*	
	Ganesh-NOM	study-CPM-CONT-PST.AP	**that time**	I-NOM	cook-PST-1.SG.M/F	

'I cooked when Ganesh was studying'

(iii) Discourse Connector

The discourse connectors are non-adverbial markers that introduce an adverbial clause to a sentence.

Example (3.168)

3.168	***āṉāl,***	*nāṉ*	*va-ra.v-illay*
	but	I-NOM	come-INF-NEG
	‘But, I am not coming’		

List of SCONJs

ataṉāl, ataṟku, āṉāl, itaṉāl, iruntapōtilum, iruppiṉum, eṉa, eṉavē, eṉum, eṉpataṟku, eṉpatāl, eṉpatil, eṉpatu, eṉpatē, eṉpatay, eṉṟa, eṉṟāl, eṉṟu, eṉṟum, ēṉeṉil, ēṉeṉṟāl, kāṭṭilum, tavi, piṉṉar, pōṉṟa, pōla, pōlavē, pōl, mēlum, etc.

3.3.3 Other

Others include PUNCT and SYM which are signs and not words.

3.3.3.1 PUNCT: punctuation

Punctuation denotes non-alphabetical/non-numeric characters and character groups. Speech corpora has symbols that represent pauses, laughter and other sounds are treated as punctuations.

Listed items using *(•, ›)* are punctuations and not SYM.

Example (3.169)

3.169	**“**	**PUNCT**
	Āhā ‘wow’	INTJ
	!	**PUNCT**
	eṉṉa ‘what’	NOUN
	ruci ‘taste’	NOUN

	”	**PUNCT**
	en̲ru ‘COMP’	SCONJ
	co-n̲n̲-ān̲ ‘tell-PST-3.SG.M’	VERB
	.	**PUNCT**

List: .(period), () (parentheses), , (comma), “ (open double quote), ” (close double quote), ; (semi-colon), : (colon), ? (question mark), ! (exclamation mark), ‘ (open single quote), ’ (close single quote), etc.

3.3.3.2 SYM: symbol

A symbol is a special entity that is not a part of any word/ number class. Most of the symbols are special non-alphanumeric characters like punctuations. Punctuations can be replaced by another word but symbols cannot be. Mathematical operators, emoticons and emoji. Etc. are grouped under SYM.

Example (3.170)

nān̲ 90% matippeṇ vāṅkiṉēn̲. ‘I got 90% mark’

3.170	*nān̲* ‘I-NOM’	PRON
	90 ‘90’	NUM
	%	**SYM**
	matippeṇ ‘mark’	NOUN
	vāṅk-iṉ-ēn̲ ‘get-PST-1.SG.M/F’	VERB
	.	PUNCT

List of SYMs

- Mathematical functors: *$, %, §,* ©
- Mathematical operators: +, −, ×, ÷, =, <, >
- Emoticons: :), ♥‿♥, 😝
- Websites/mail ids: *www.int.org , tulips9@gmail.com*

3.4 Multi-token word expander

Tamil is an agglutinative language which encodes multiple information in a single word. Multi-token words cannot be directly fed into the POS module as they fail to identify the correct tag. The module multi-token word expander works before the data is sent to POS and morph tagging. The following rules are followed in multi- token word expander, which would split words with multiple syntactic information into different words.

(i) determiner+noun

When a determiner occurs with a noun as a single word, the word is split into two different units.

Example (3.171) *ippakuti= ip+pakuti* 'This place'

Example (3.172) *akkaray= ak+karay*'That bank of the river'

(ii) noun+verb

Noun and verb combinations are split into different words.

Example (3.173) *iṭamākum= iṭam+ākum* 'place+copula'

Example (3.174) *puka[per̲r̲atu= puka[+pe̲r̲r̲atu* 'fame+got'

(iii) verb+auxiliary (+auxiliary)

One or two auxiliaries attached to the verb are split.

Example (3.175) *nirampiyul̲l̲atu= nirampi+ul̲l̲atu* 'getting full'

Example (3.176) *nirappappaṭṭul̲l̲atu= nirappa+paṭṭu+ul̲l̲atu* 'it is being filled'

(iv) verb+particle

Combination of verb and particle are split into two different words.

Example (3.177) *ceytavaray= ceyta+varay* 'till what has been done'

Example (3.178) *pārkkumpōtu= pārkkum+pōtu* 'while seeing'

(v) number+verb

When a number and a verb or copula occurs together as a single unit, it has to be split.

Example (3.179) *pattākum= pattu+ākum* 'ten+copula'

Example (3.180) *iraṇṭākum= iraṇṭu+ākum* two+copula'

(vii) Noun+verb+auxiliary

Nouns followed by verbs and auxiliaries occur together, it has to be split.

Example (3.181) *kāyamaṭayyavillay= kāyam+aṭayya+illay* 'didn't get wound'

Example (3.182) *maṟucīramaykkappaṭṭa= maṟucīr+amaykkap+paṭṭa* 'being renovated'

(ix)Number+noun

The unit of number and noun need a split.

Example (3.183) *orunāḷaykku= oru+nāḷaykku* 'for one day'

(xi) Adverb+adjective

Adverb and adjective combination requires a split.

Example (3.184) *mikacciṟanta= mikac+ciṟanta* 'very good'

(xiii) noun/pronoun+Clitics

Occurrence of clitics with a noun or pronoun requires a split.

Example (3.185) *atutāṉ= atu+tāṉ* 'that+emphasis'

Example (3.186) *iṭamē= iṭam+ē* 'place+emphasis'

(xv) Pronoun+adposition

Pronoun adposition combination requires a split.

Example (3.187) *itayviṭa= itay+viṭa* 'than this'

(xvii) unconventional combination

Unconventional/ unusual combinations which do not give a meaning needs a split. Such units are found due to tokenisation errors.

Example (3.188) *kōṭṭayintiyaṉ= kōṭṭay+intiyaṉ* ‘fort+Indian’

Example (3.189) *5veḻināṭṭavar= 5+veḻināṭṭavar* ‘5+ foreigners’

Certain combinations of categories are considered as a single unit. These combinations are not required to split.

(i) Noun+Noun

Example (3.190) *marakkaṭṭ*ay= **maram+*kaṭṭay* ‘wood+rod’

Example (3.191) *ōṭṭuvīṭu= *ōṭṭu+*vīṭu* ‘thatch roof+house’

(ii) Adposition+Noun

Example (3.192) *kīḻkkōvil= *kīḻ+*kōvil* ‘down+temple’

(iii) number+noun

Example (3.193) *irupuṟam= *iru+*puṟam* ‘two+sides’

(iv) adjective+noun

Example (3.194) *putuvoḻi= *putu+*oḻi* ‘new light’

Example (3.195) *nīlaniṟa= *nīla+*niṟa* ‘blue colour’

(v) Noun+adverb

Example (3.196) *vīriyamikkatāka= *vīriyam+*mikkatāka* ‘powerful’

(vi) proper noun+noun

Example (3.197) *iñciccāṟu= *iñci+*cāṟu* 'ginger juice’

(v) noun + verb (frozen forms)

Frozen forms of words should never be split.

Example (3.198) *āṭcipuri= *āṭci+*puri* ‘to rule’

Chapter 4

Domain Specific Syntactic Treebank

The grammatical relationship between the words in a sentence has a unique classification in Universal Dependency tagset. It has a space for language specific features. All the dependents of the head, and the functional words with their heads are connected to the head. The head is not always the verb of the sentence in Tamil as seen in example (38). The relation 'dep' is used when none of the listed relations are possible. The language specific components are written after a colon, following the main classification as in 'obl:tmod'. The table 4.1 shows the organization of grammatical categories. This section describes each of these tags with examples as follows.

	Nominals	**Clauses**	**Modifier words**	**Function words**
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative dislocated	advcl	advmod discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
cconj cc	fixed flat compound	list parataxis	orphan goeswith	punct root dep

Table 4.1: organized list of UD syntactic tags (www.universaldependencies.org)

4.1 Core arguments

Core argument is a functional category, sorted under the structural categories, nominals and clauses. It deals with the clausal predicates of a sentence.

4.1.1 Nominals

Nominals include nsubj, obj and iobj, which denote nouns.

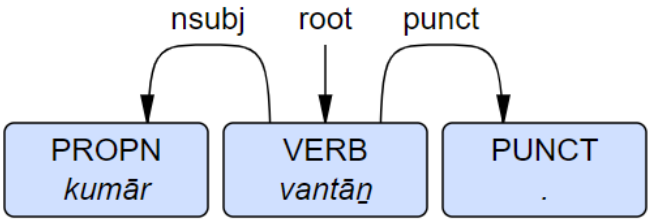
4.1.1.1 nsubj: nominal subject

The syntactic subject of the clause, belonging to the nominal core argument is 'nsubj'. It passes the grammatical test for subject and since it acts as the do-er of the action, it is the proto-agent of the clause.

Example (4.1)

#text = ***kumār*** *vantāṉ*.

#trans = '**Kumar** came'



The tag 'nsubj' occurs with multiple case markers as seen below:

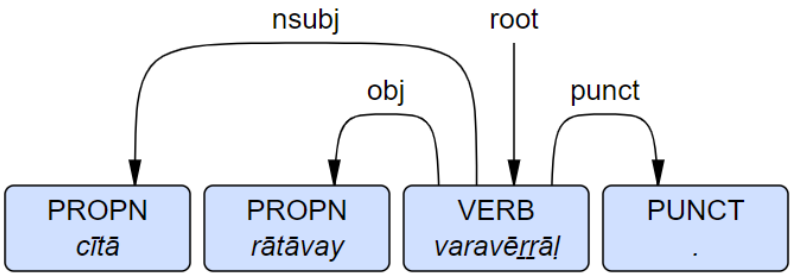
(i) nsubj in nominative case

Generally, 'nsubj' occurs in the nominative case in Tamil. Nominative case marker in Tamil is ∅ and thus, the noun remains the same without any additions.

Example (4.2)

text = ***cītā*** *rātāvay varavēṟṟāḷ*.

trans = '**Sita** welcomed Radha'



'nsubj' is found with different case markers in Tamil as certain predicates require their 'nsubj' to be case-marked by non-nominative case markers such as the dative, locative and instrumental markers.

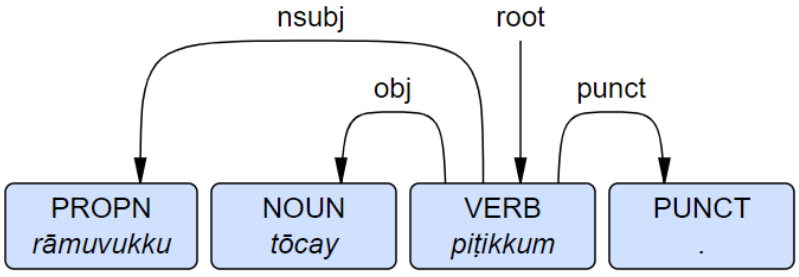
(ii) nsubj in dative case (nsubj:nc)

The dative marked subject (non-canonical subject) acts as an 'experiencer' subject as the verb agrees with the object. In Tamil, stative predicates expressing the notion of mental, emotional and physical experience require the case marking pattern of DAT-ACC (Lehmann, 1993:180).

Example (4.3)

#text = ***rāmuvukku*** *tōcay piṭikkum* .

#trans = '**Ramu** likes Dosa'



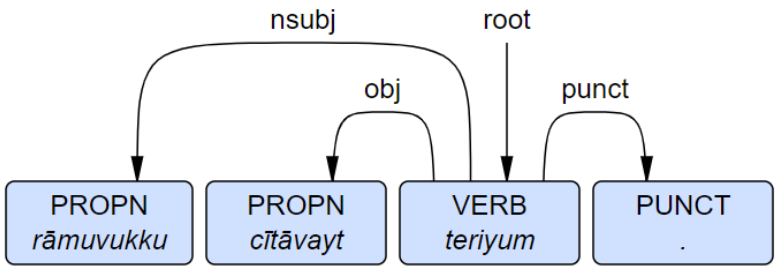
(a) Verbs of mental experience

When the verbs (as root of the sentence) such as *teri* 'know' and *puri* 'understand' occur in Tamil, the *logical* subject is marked with the dative case marker. There is a default subject- verb agreement in such cases.

Example (4.4)

#text = ***rāmuvukku*** *cītāvayt teriyum* .

#trans = '**Ramu** knows Sita'



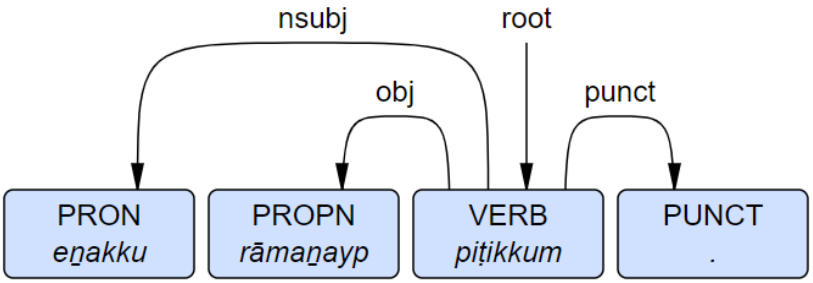
(b) Verbs of emotional experience

Verbs like *piti* 'like' etc., in Tamil express emotional experience with the dative-marked subject.

Example (4.5)

#text = ***eṉakku*** *rāmaṉayp piṭikkum* .

#trans = 'I like Raman'



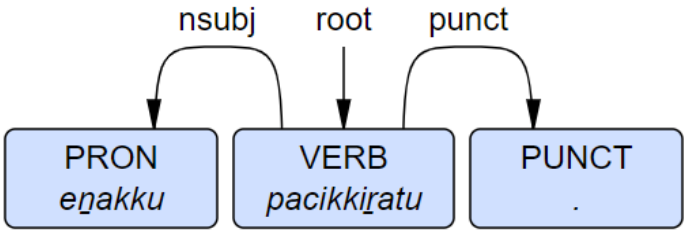
(c) Verbs of psycho-semantic, physical and physiological experiences

Verbs such as *paci* ‘be hungry’, *vali* ‘be painful’, and *ari* ‘be itching’ in Tamil conveys psycho-semantic, physical and physiological experiences which requires their subject with the dative marker.

Example (4.6)

#text = ***eṉakku*** *pacikkiṟatu* .

#trans = ‘**I** am hungry’



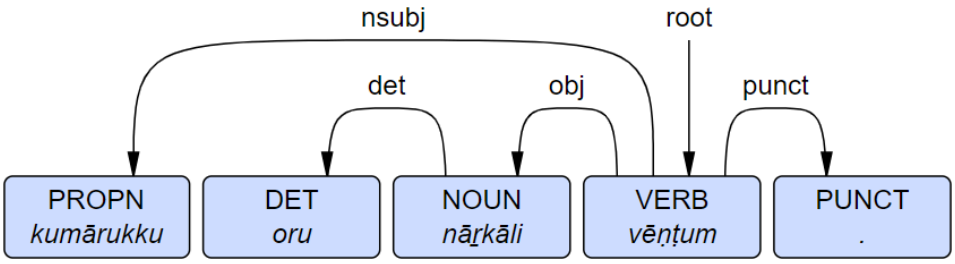
(d) Modal Auxiliary *'vēṇṭum'*

Auxiliaries like *vēṇṭum* ‘want’ in Tamil require the subject in the dative case marker.

Example (4.7)

#text = ***kumārukku*** *oru nāṟkāli vēṇṭum* .

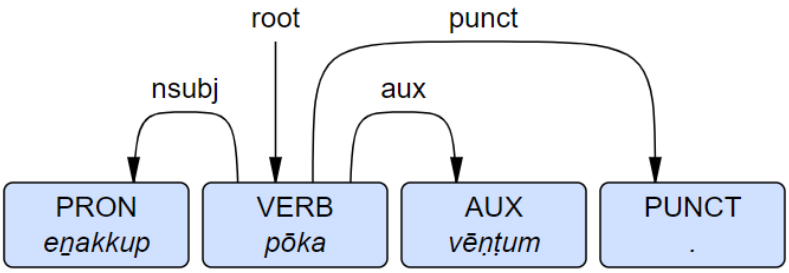
#trans = **Kumar** wants a chair.



Example (4.8)

#text = ***eṉakkup*** *pōka vēṇṭum.*

#trans = ‘**I** wanted to go’



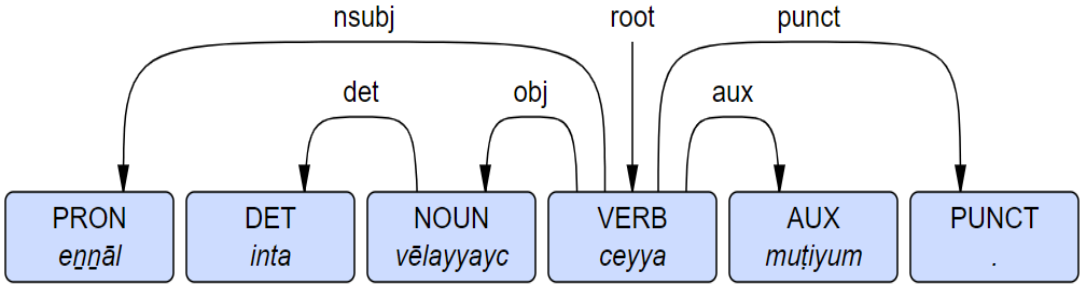
(iii) nsubj in instrumental case (nsubj:nc)

When the predicate indicates the capabilitative mood, the subject is optionally marked for the instrumental case and the verb gets default agreement (third person-neuter).

Example (4.9)

#text = ***eṉṉāl*** *inta vēlayyayc ceyya muṭiyum .*

#trans = **'I** can do this work'



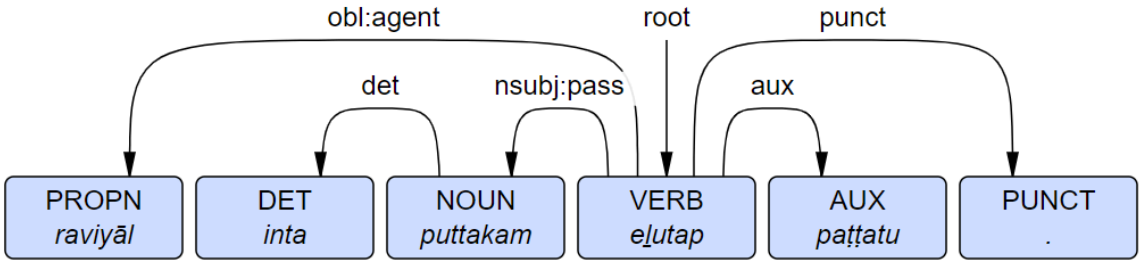
(iv) nsubj:pass: nsubj in passive constructions

The syntactic subject of the passive clause is a language- specific tag denoted by 'nsubj:pass'.

Example (4.10)

#text = *raviyāl inta* ***puttakam*** *eḻutappaṭṭatu.*

#trans = 'The **book** was written by Ravi'



Test case for 'nsubj'

nsubj

(i) NOUN/PRON/PROPN are subjects

(ii) NOUN, 0-marking

iii) NOUN GNP (agreement) = VERB GNP

nsubj:pass

(i) NOUN/ PRON/PROPN are subjects

(ii) NOUN , 0-marking

(iii) NOUN GNP (agreement) = VERB GNP

(iv) Verb-*paṭu*

'nsubj' in dative and instrumental case construction

(i) NOUN/ PRON/PROPN are subjects

(ii) NOUN , -*ku* marking (dative); -*āl* marking (instrumental)

(iii) NOUN GNP (agreement) ≠ VERB GNP

(iv) object of the same sentence = VERB GNP

4.1.1.2 obj: direct object

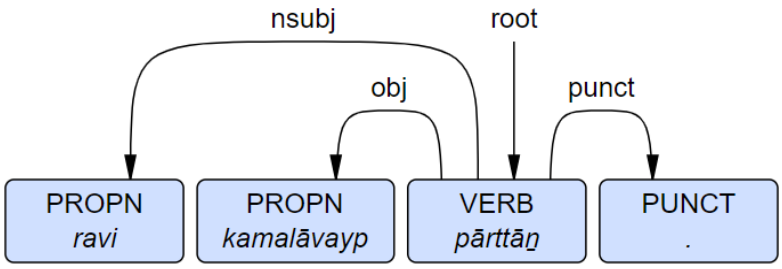
The most important core argument nominal which is not the subject of the sentence is 'obj'. The noun/NP which undergoes a change of state or motion or which becomes the most affected participant (proto-patient) is 'obj'.

In Tamil, the rational objects are marked by the accusative case marker '*ay*' explicitly, and irrational objects are optionally marked by the accusative case.

Example (4.11)

#text = *ravi* ***kamalāvayp*** *pārttāṉ*.

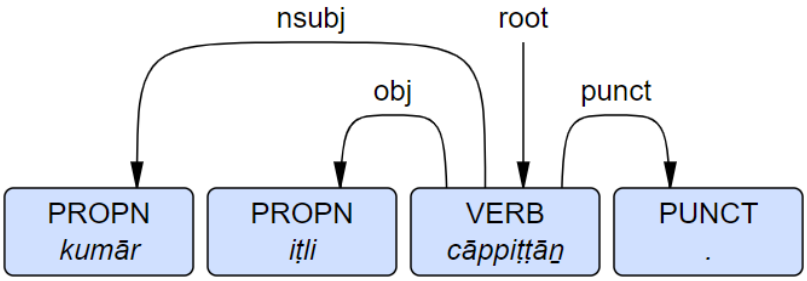
#trans = 'Ravi saw **Kamala**'



Example (4.12)

#text = *kumār* ***iṭli*** *cāppiṭṭāṉ* .

#trans = 'Kumar ate **idli**.



Test case for 'obj'

(i) NOUN/ PRON/PROPN are direct objects

(ii) NOUN , -*ay* marking (for animate) or 0-marking (for inanimate)

(iii) verb should be transitive

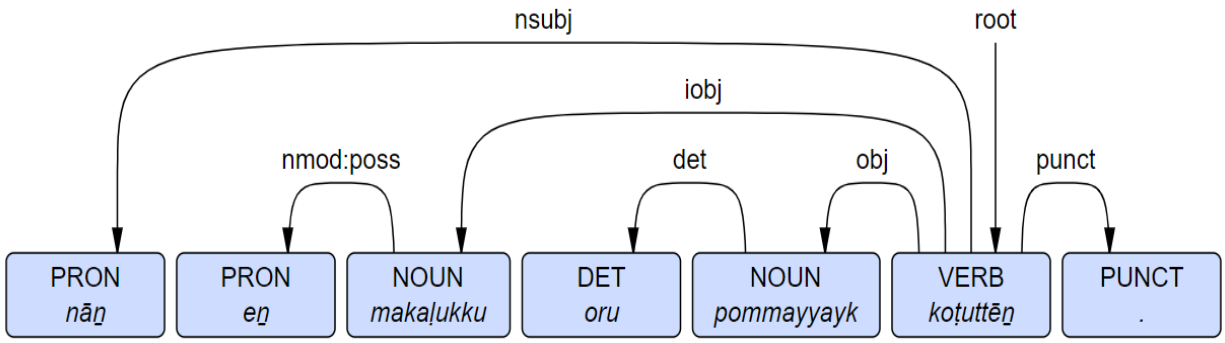
4.1.1.3 iobj: indirect object

The noun phrase which is the recipient of a ditransitive verb i.e. indirect object is marked as 'iobj'. When there are two verbs in a sentence, the least affected object is termed 'iobj'. It belongs to the core argument of the sentence and it is marked with dative case marker *-ku* in Tamil.

Example (4.13)

#text = *nāṉ eṉ* ***makaḷukku*** *oru pommayyayk koṭuttēṉ.*

#trans = 'I gave a toy to my **daughter'**



Test case for 'iobj'

(i) NOUN/ PRON/PROPN are indirect objects

(ii) NOUN , *-ku/-iṭam* marking

(iii) verb should be ditransitive

4.1.2 Clauses

At the clausal level, the syntactic tags csubj, ccomp and xcomp are used.

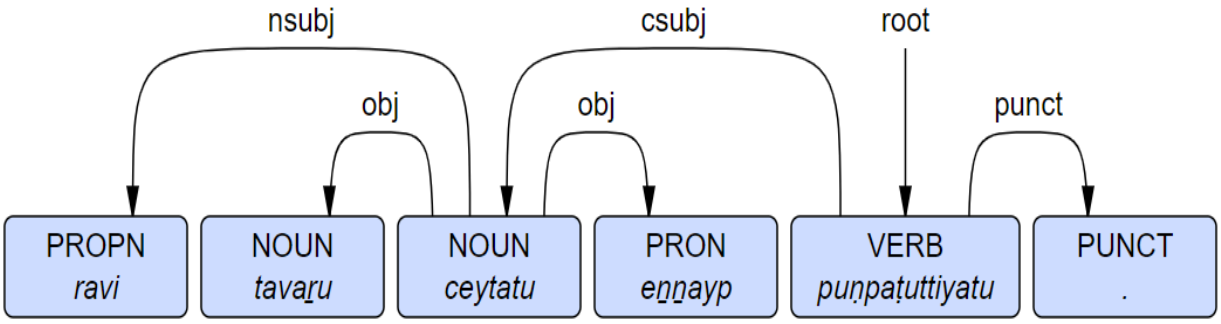
4.1.2.1 csubj: clausal subject

When the syntactic subject of a predicate is realized as a whole clause, it is marked as 'csubj'. The root of the sentence can either be a verb or a noun (in case of copula construction).

Example (4.14)

#text = *ravi tavaṟu* ***ceytatu*** *eṉṉayp puṇpaṭuttiyatu.*

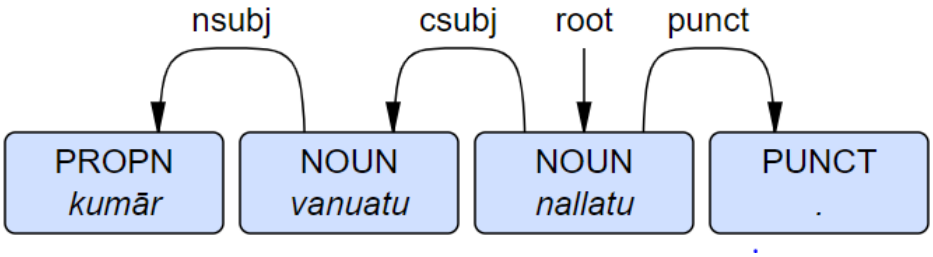
#trans = 'Ravi's wrong **deed** hurted me'.



Example (4.15)

#text= *kumār* ***vanuatu*** *nallatu.*

#trans= 'Kumar's **arrival** was good'



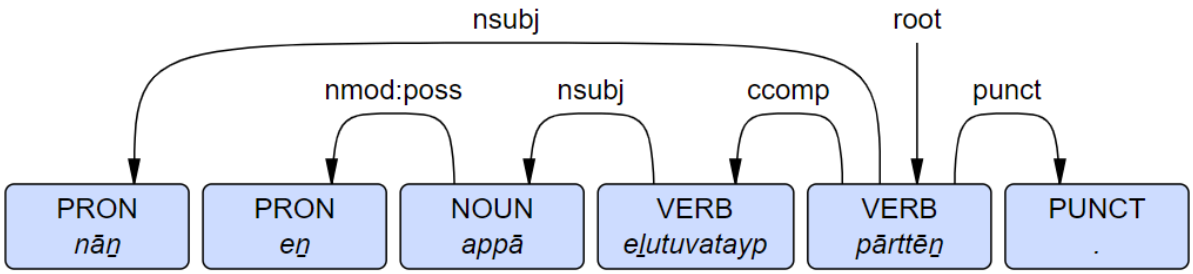
4.1.2.2 ccomp: clausal complement

Clausal complement is a dependent clause which is marked when a clause functions like an object.

Example (4.16)

#text = *nāṉ eṉ appā* ***eḻutuvatayp*** *pārttēṉ.*

#trans = 'I saw my dad **writing'**.

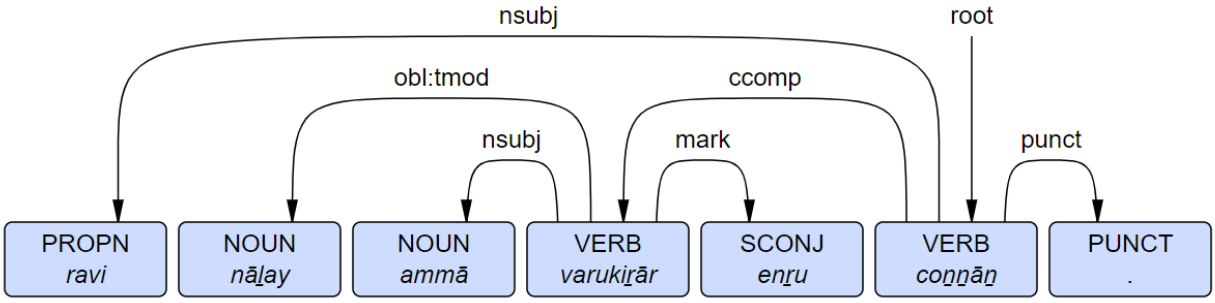


There is no complementizer being used in the above example. The next example is with the complementizer illustrated below:

Example (4.17)

#text= *ravi nāḻay ammā* ***varukiṟār*** *enṟu coṉṉāṉ.*

#trans= 'Ravi said that mom is **coming** tomorrow'



4.1.2.3 xcomp: open clausal complement

An open clausal complement does not have its own subject and the subject is determined by the external clause which can either be subject or object of the next higher clause.

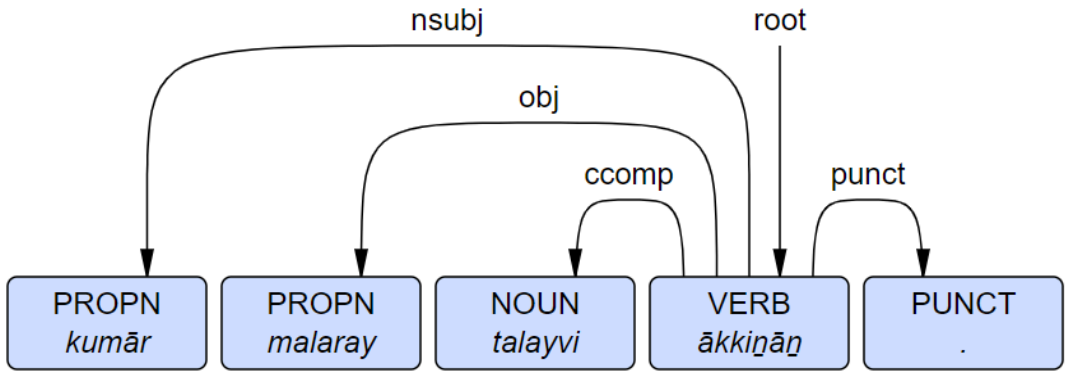
(i) 'xcomp': a nominal complement

The verb *-ākku* 'to become' has a special quality of marking nominal complements as 'xcomp'.

Example (4.18)

#text= *kumār malaray* ***talayvi*** *ākkiṉāṉ*

#trans= ‘Kumar made Malar a **leader’**



(ii) ‘xcomp’: a verbal complement

When ‘xcomp’ occurs as a verbal complement, it can either be subject controlled or object controlled.

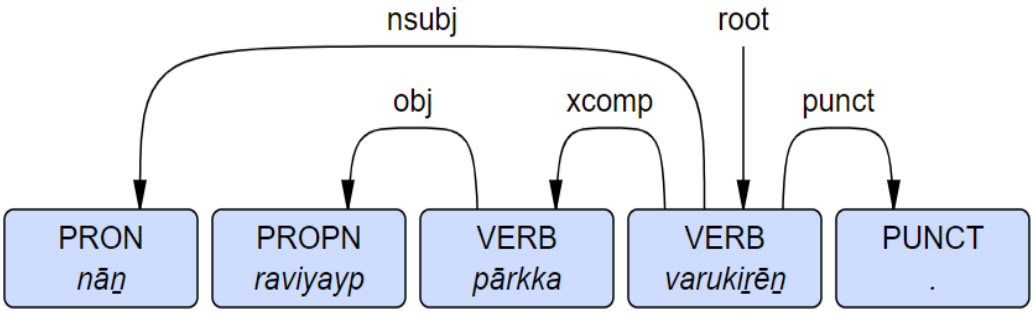
(a) Subject control

The subject of the next higher clause takes control of the infinitive verb in Subject controlled ‘xcomp’.

Example (4.19)

#text= *nāṉ raviyayp **pārkka** varukiṟēṉ*

#trans= ‘I am coming **to see** Ravi’



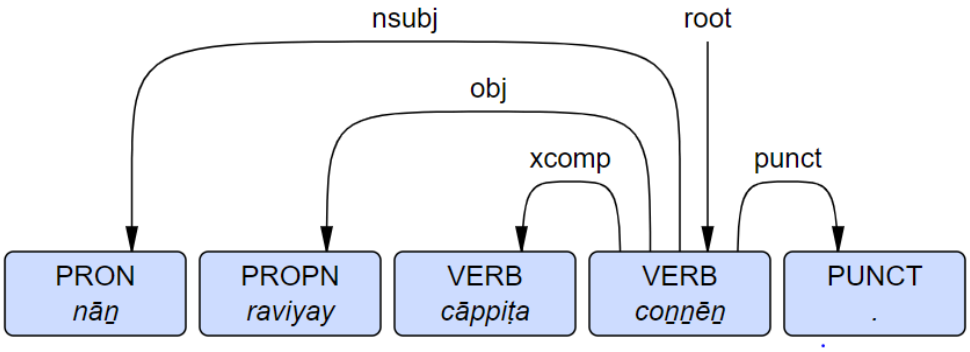
(b) Object control

The object of the next higher clause takes control of the infinitive verb in Object controlled 'xcomp'.

Example (4.20)

#text= *nāṉ raviyay* ***cāppiṭa*** *coṉṉēṉ*

#trans= 'I asked Ravi **to eat**'



4.2 Non-core dependents

Non-core dependents are grouped under the structural categories- Nominals, Clauses, Modifier words and Function words. The following syntactic tags are found under the non-core dependents.

4.2.1 Nominals

The tags obl, vocative, expl and dislocated are grouped under Nominal non-core dependents

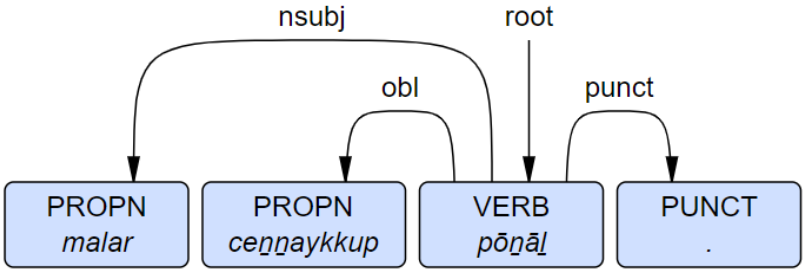
4.2.1.1 obl: oblique nominal

When a nominal (noun, pronoun, or NP) acts as an adjunct or non-core (oblique) argument, the 'obl' relation is used.

Example (4.21)

#text= *malar* ***ceṉṉaykkup*** *pōṉāḷ*

#trans= ‘Malar went to **Chennai’**



Language-specific tags of ‘obl’

The tag obl is used for a nominals with various markers and thus, it is classified into language-specific tags according to syntactic cues as seen below:

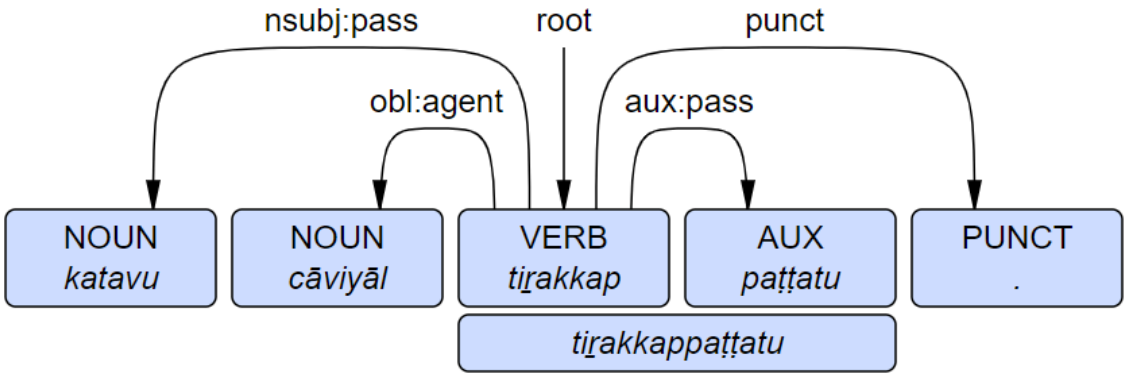
(i) obl: agent: oblique nominal agent

In passive constructions, the language specific tag, obl:agent is used for nouns which are agents of passivized verbs.

Example (4.22)

#text= *katavu* ***cāviyāl*** *tiṟakkappaṭṭatu*

#trans= ‘The door was opened by the **key’**



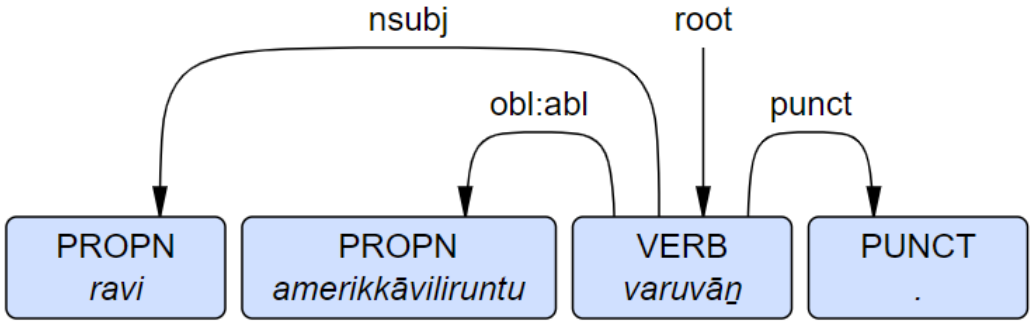
(ii) obl:abl: oblique nominal ablative

The noun referring to source is marked by the relation ‘obl:abl’. The Tamil word final marker for such nouns is -*iliruntu.*

Example (4.23)

#text= *ravi* ***amerikkāviliruntu*** *varuvāṉ*

#trans= ‘Ravi will come from **America’**



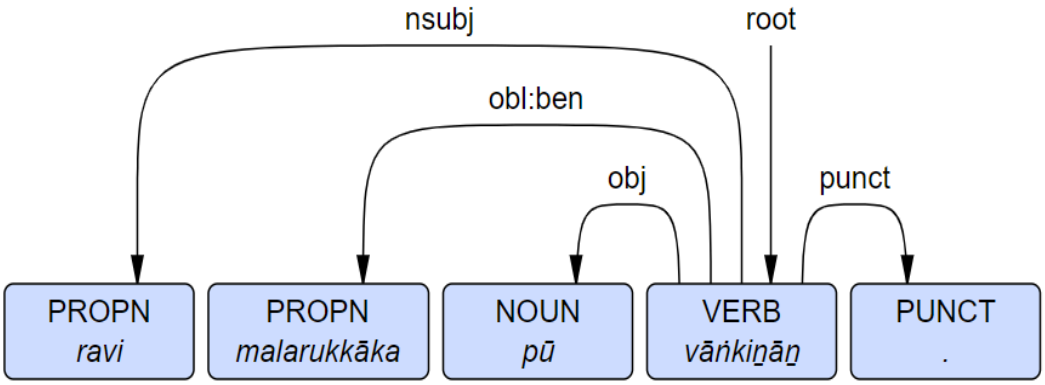
(iii) obl:ben: oblique nominal benefactive

The beneficiaries of the action performed are marked with the relation ‘obl:ben’. The suffix -(*u)kkāka* is observed as the benefactive marker in Tamil.

Example (4.24)

#text= *ravi* ***malarukkāka*** *pū vāṅkiṉāṉ*

#trans= ‘Ravi bought flowers for **Malar’**



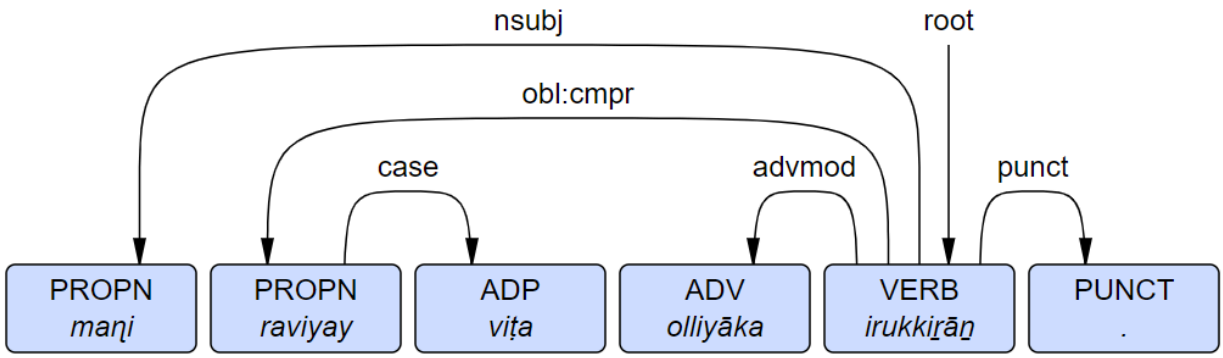
(iv) obl:cmpr: oblique nominal comparison

When two nominals with the relation ‘obl’ are compared, the language specific tag ‘obl:cmpr’ is used.

Example (4.25)

#text= *maṉi* ***raviyay*** *viṭa olliyāka irukkiṟāṉ*

#trans= ‘Mani is thinner than **Ravi’**

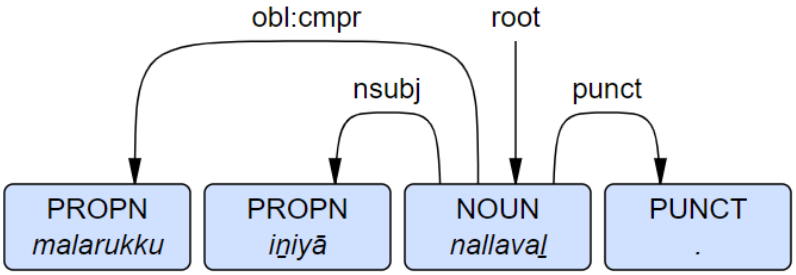


Comparison can be made even without using the ADP *viṭa* as seen below:

Example (4.26)

#text= ***malarukku*** *iṉiyā nallavaḷ*

#trans= ‘Iniya is better than **Malar’**



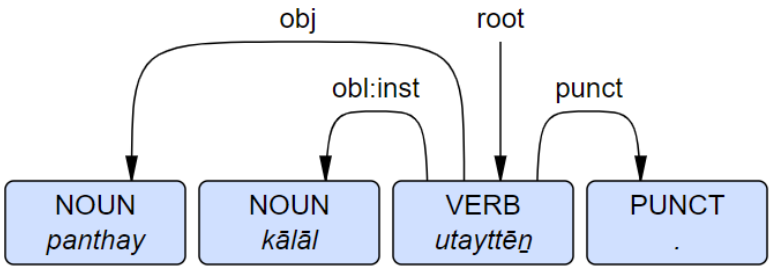
(v) obl:inst: oblique nominal instrumental

Nominals which act as instruments for the actions performed in the text are marked with the relation ‘obl:inst’. *āl* is the instrumental case marker seen in such cases.

Example (4.27)

#text= *pantay* ***kālāl*** *utayttēn̲*

#trans= '(I) kicked the ball with the **leg'**



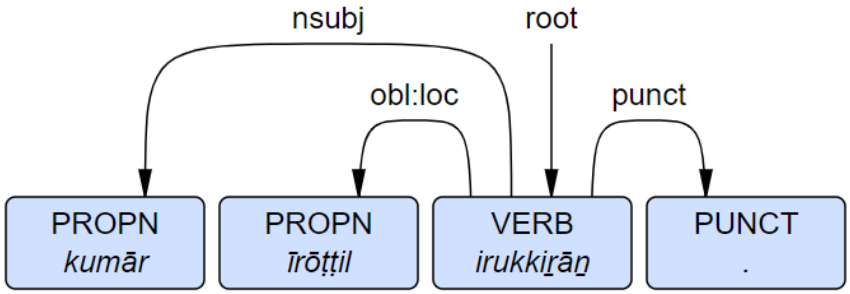
(vi) obl:loc:oblique nominal location

Nominals which denote location of another nominal are given the language-specific relation 'obl:loc'. Locative marker in Tamil is *-il.*

Example (4.28)

#text= *kumār* ***īrōṭṭil*** *irukkiṟān̲*

#trans= 'Kumar is in **Erode'**



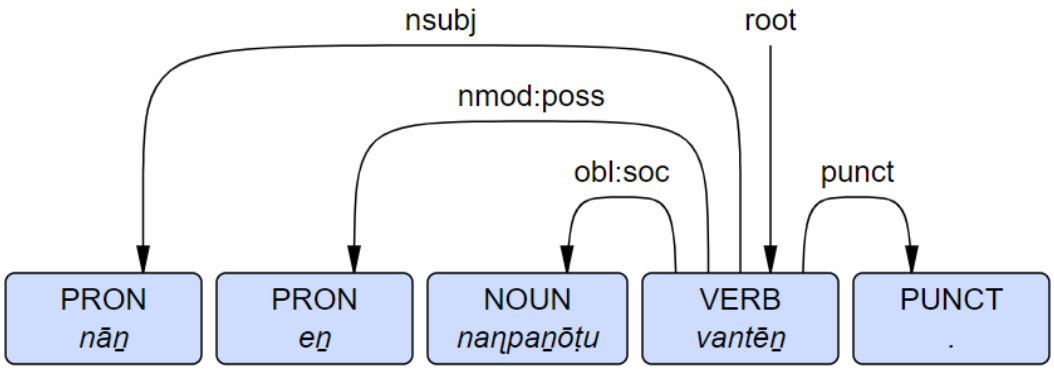
(vii) obl:soc: oblique nominal sociative

Associative/ sociative case in Tamil is expressed by the marker *-ōṭu.* The relation 'obl:soc' is used for such nouns.

Example (4.29)

#text= *nān̲ en̲* ***naŋpan̲ōṭu*** *vantēn̲*

#trans= ‘I came with my **friend’**



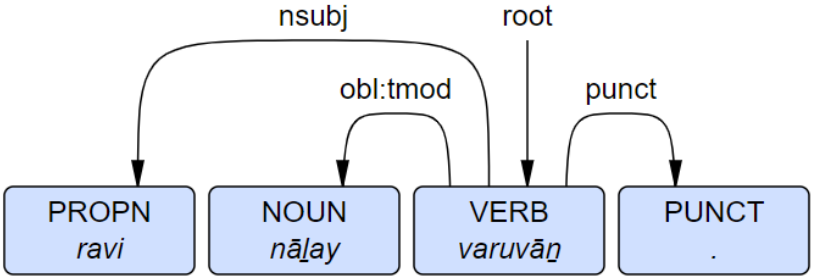
(viii) obl:tmod: oblique nominal temporal modifier

The oblique form of noun which specifies time is denoted by the relation ‘obl:tmod’.

Example (4.30)

#text= *ravi* ***nāḷay*** *varuvāṉ*

#trans= ‘Ravi will come **tomorrow’**



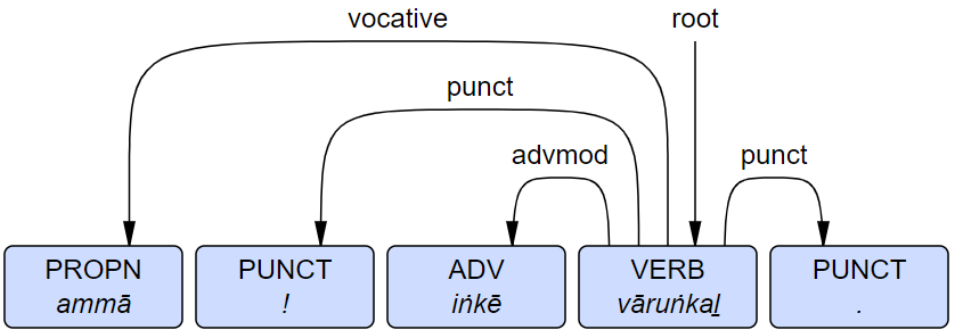
4.2.1.2 vocative: vocative

When addressing a dialogue participant in a text, the ‘vocative’ relation is used.This usage is common in discussions, dialogue, emails, newsgroup postings, etc. The relation connects the host sentence to the addressee's name. This tag is seen mostly in the conversation data set.

Example (4.31)

#text= ***ammā!*** *iṅkē vāruṅkaḷ*

#trans= ‘Mom! Come here’



4.2.1.3 expl: expletive

Nominals that are pleonastic or expletives are captured by this relation. These are nominals that show up in a predicate's argument position but do not fulfill any of the predicate's semantic responsibilities. The governor is the primary predicate of the phrase, which can be either a verb, predicate adjective, or predicate noun. Such an occurrence is not found in annotated Tamil data.

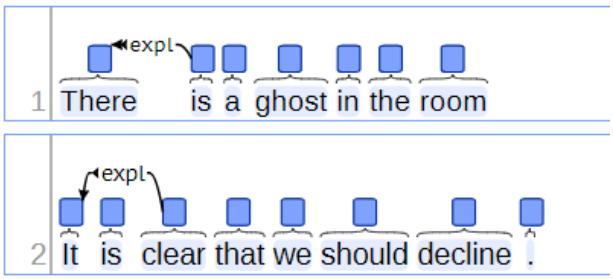


Figure 4.1: Example for ‘expl’ (www.universaldependencies.org)

4.2.1.4 dislocated: dislocated elements

The fronted or postponed elements that do not have any grammatical relations to the head of the sentence is termed ‘dislocated’. These elements are mostly found in the sentence's final position, optionally separated by comma. This tag is not found in the annotated data set.

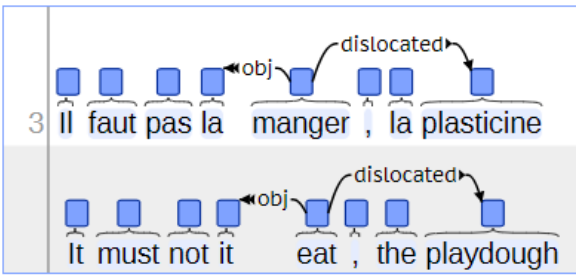


Figure 4.2: Example for 'dislocated' (www.universaldependencies.org)

4.2.2 Clauses

Adverbial clauses are documented under clauses, which is considered a non-core element of the sentence.

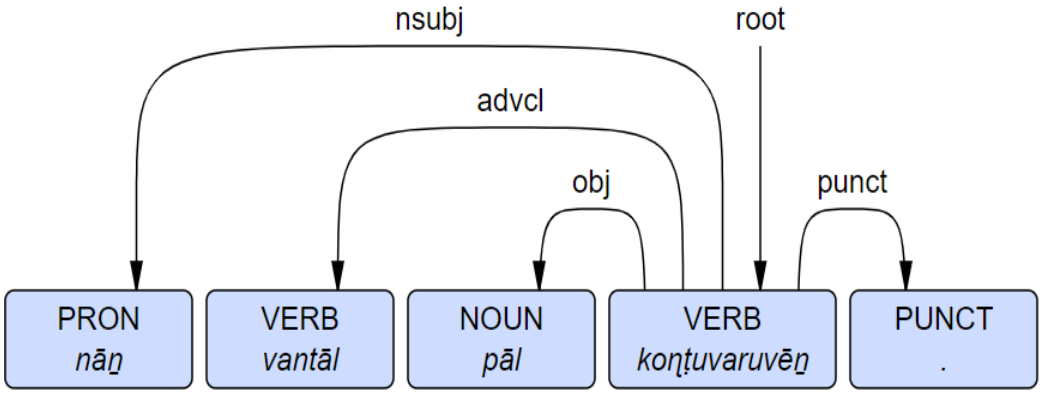
4.2.2.1 advcl: adverbial clause modifier

An adverbial clause modifier (advcl) is a clause which includes temporal clause, consequence, conditional clause, purpose clause, etc., modifying a verb or other predicate, as a modifier. It is considered an adjunct of the sentence. The dependent of advcl is any clause, a part of the main predicate of the sentence.

Example (4.32)

#text= *nāṉ* ***vantāl*** *pāl koṇṭuvaruvēṉ*

#trans= 'I will bring milk if I **come'**



4.2.3 Modifier words

'advmod' and 'discourse' are the two tags being classified under modifier words in non-core dependents.

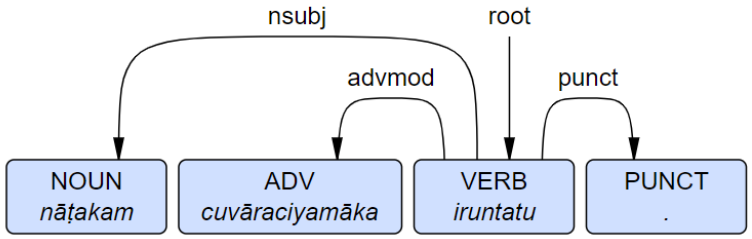
4.2.3.1 advmod: adverbial modifier

An adverbial modifier (advmod) is a modifying word which serves to modify a verb/ an adjective/ another adverb or nouns of space and time. All *-āka* suffixes are tagged POS 'adv' and dependency relation 'advmod' in Tamil data.

Example (4.33)

#text= *nāṭakam* ***cuvāraciyamāka*** *iruntatu*

#trans= 'The series was **interesting'**



Language - specific tags

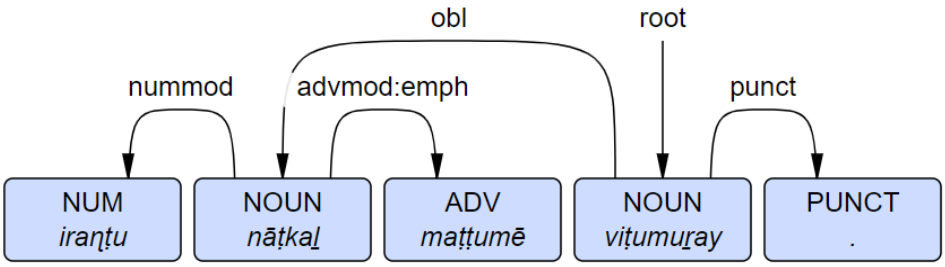
(i) advmod:emph

Some adverbs can also modify nouns (e.g., ***only*** *on Monday*). The subtype of 'advmod' has to be used

Example (4.34)

#text= *iraṇṭu nāṭkaḷ* ***maṭṭumē*** *viṭumuṟay*

#trans= **'Only** two days are holidays'



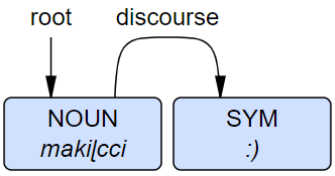
4.2.3.2 discourse: discourse element

The interjections and other discourse particles and elements like smilies, which are not directly related to the grammatical relations of the sentence are marked by the tag ‘discourse’. It expresses the emotions of the text/ sentence.

Example (4.35)

#text= *makiḻcci* ***:)***

#trans= ‘Happy **:)**’



4.2.4 Function words

The non-core dependency relations ‘aux’, ‘cop’ and ‘mark’ are listed under function words.

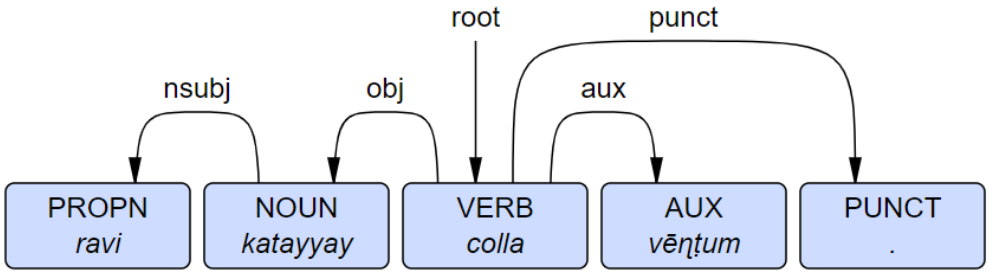
4.2.4.1 aux: auxiliary

Tense, mood, aspect, voice or evidentiality are the functions expressed by the dependency relation ‘aux’.

Example (4.36)

#text= *ravi katayyay colla* ***vēṇṭum***

#trans= ‘Ravi **should** tell the story’



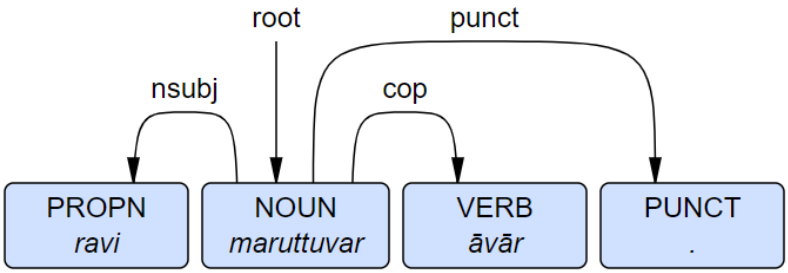
4.2.4.2 cop: copula

Copula verbs are optional in Tamil. It is limited in usage. Copula links subject to a non-verbal predicate.

Example (4.37)

#text= *ravi maruttuvar* ***āvār***

#trans= 'Ravi **is** a doctor'



The head of **'**nsubj' is not always a verb in Tamil. When the nominal and adjectival predicates occur optionally with the copula verb '*āku*', non-verbal predicates are considered as head (root) and the copula verb if present, it is related as aux (auxiliary) to the root. So, in the above example, *maruttuvar* 'doctor' is the root of the sentence.

4.2.4.3 mark: marker

A marker is the word marking a clause as subordinate to another clause.

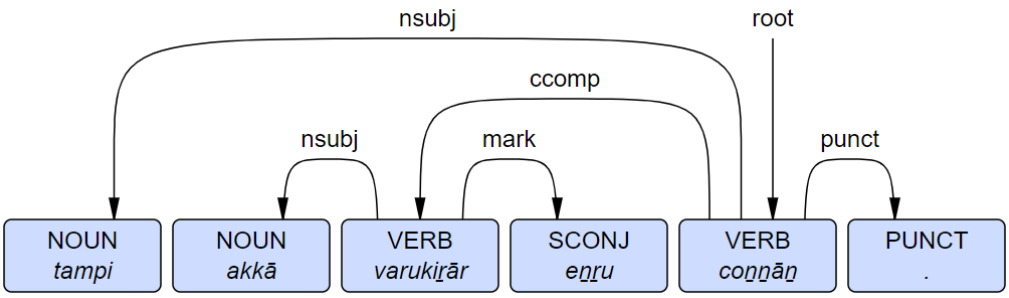
- Words like *e̲na/e̲n̲ru* 'that' are occurrences in complement clauses and relative clauses
- The clitics like *-um/-ā/ē* that are separated from the word are marked by the relation 'mark'

- The marker is a dependent of the subordinate clause head.

Example (4.38)

#text= *tampi akkā varukiṟār **enru** coṉṉāṉ*

#trans= ‘Brother said **that** the sister is coming’



4.3 Nominal dependents

Nominal dependents are classified into nominals, clauses and modifier words, which are dependents of nominals.

4.3.1 Nominals

Nominals in this category include ‘nmod’,’appos’ and ‘nummod’.

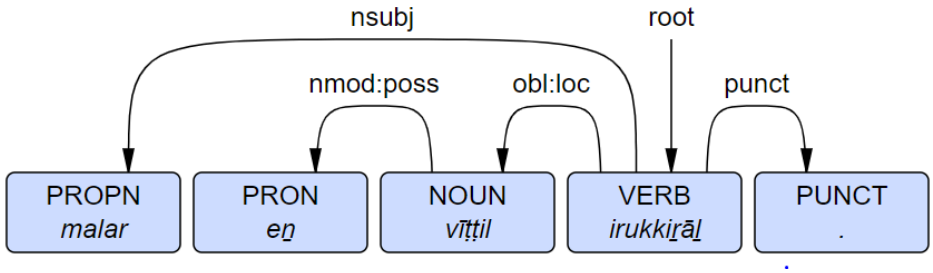
4.3.1.1 nmod: nominal modifier

Nominal dependents of another noun or NP functioning as an attribute or possession are classified under ‘nmod’. nmod:poss (possessive nominal modifier) is used for a nominal modifier that occurs before its head in the specifier position in an oblique or possessive marker. ‘nmod:poss’ is the commonly used tag in Tamil for possession.

Example (4.39)

#text= *malar **en** vīṭṭil irukkiṟāḷ*

#trans= ‘Malar is in **my** home’

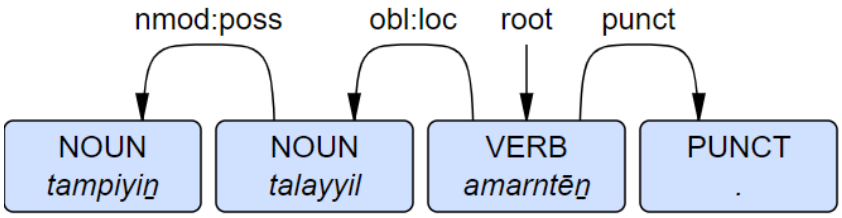


Possession can also be expressed by markers such as *-iṉ, uṭayya* in Tamil.

Example (4.40)

#text= ***tampiyiṉ/uṭayya*** *talayyil amarntēṉ*

#trans= '(I) sat on **brother's** head



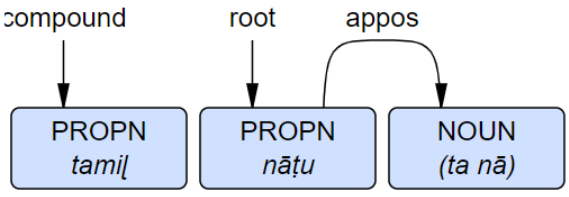
4.3.1.2 appos: appositional modifier

'appos' is a nominal which defines, modifies, names, or describes the previous nominal. It includes parenthesized examples and abbreviations as well.

Example (4.41)

#text= tamiḻ nāṭu (**ta nā**)

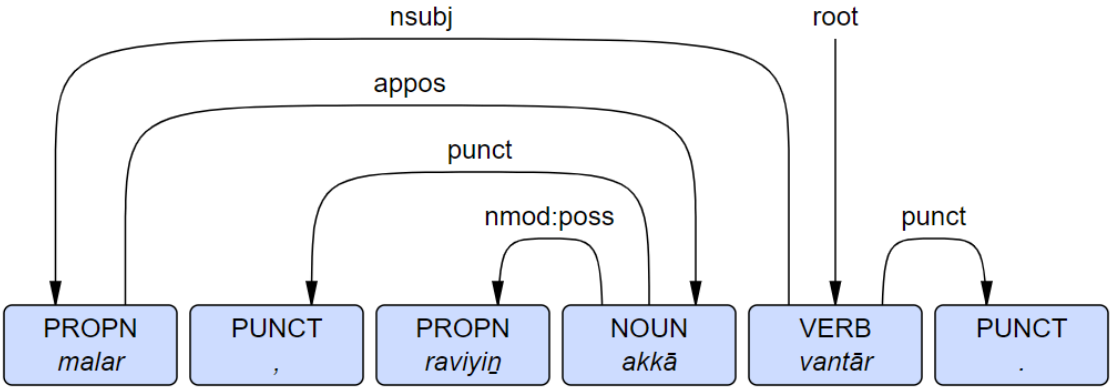
#trans= 'Tamil Nadu (**TN**)



Example (4.42)

#text= *malar, **raviyiṉ** akk*ā vantār

#trans= ‘Malar, **Ravi’s** sister came’



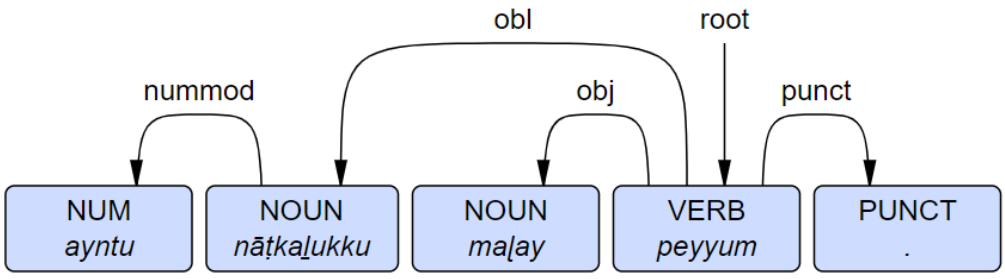
4.3.1.3 nummod: numeric modifier

The tag ‘nummod’ modifies any nominal with respect to quantity.

Example (4.43)

#text= ***ayntu** nāṭkaḻukku maḷay peyyum*

#trans= ‘It will rain for **five** days’



4.3.2 Clauses

The dependency tag ‘acl’ is a nominal dependent clause.

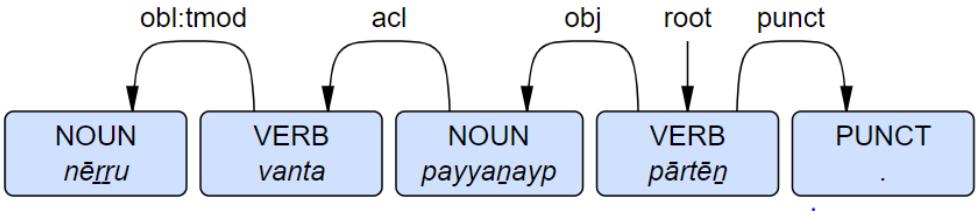
4.3.2.1 acl: clausal modifier of noun (adnominal clause)

Any clause that modifies a nominal is tagged 'acl'. 'acl' differs from 'advcl' which modifies a predicate. The head of 'acl' is a noun that is being modified, and the dependent is the head of the clause that modifies the noun.

Example (4.44)

#text= *nēr̲ru* ***vanta*** *payyaṉayp pārtēṉ*

#trans= 'I saw the boy who **came** yesterday'



4.3.3 Modifier words

The relation 'amod' is the only modifier word found in nominal dependents.

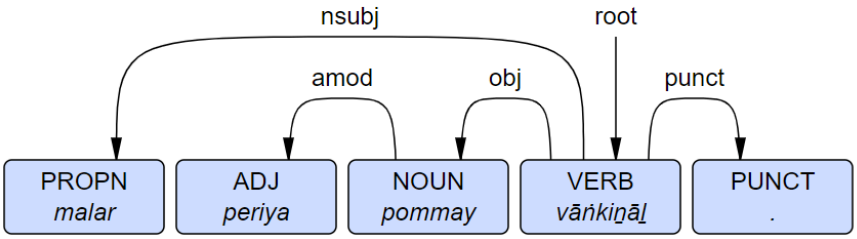
4.3.3.1 amod: adjectival modifier

'amod', is an adjectival phrase that modifies a noun/pronoun. The relation can be used in a compositional way (***periya*** *malay* 'big mountain') and idiomatic way (***paccai*** *taṉṉīr* 'cold water') as well.

Example (4.45)

#text= *malar* ***periya*** *pommay vāṅkiṉāl̲*

#trans= 'Malar bought a **big** doll'



4.3.4 Function words

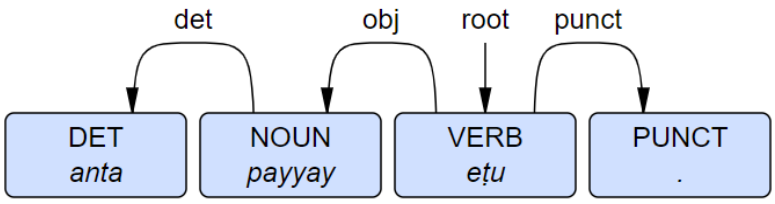
The category of function words include ‘det’, ‘clf’ and ‘case’

4.3.4.1 det: determiner

The relation determiner ‘det’ links the head of a nominal and the POS category DET. A cue for the category is that POS DET will consequently hold the syntactic relationship det to the preceding or following nominal.

Exceptions: In English, *my* is currently given the POS tag DET. But in Tamil, such possessive determiners are marked as ‘nmod’, so that it is parallel with other possessive constructions.

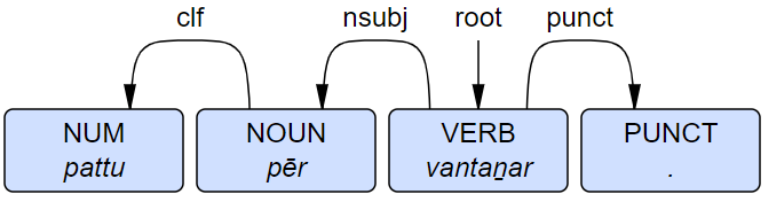
Example (4.46)
#text= ***anta*** *payyay eṭu*
#trans= ‘(You) take **that** bag’



4.3.4.2 clf: classifier

Classifiers are words which accompany nouns in particular grammatical contexts. The most common usage is numeral classifiers, in which the classifier is marked to the number, used in counting the objects.

Example (4.47)
#text= *pattu* ***pēr*** *vantaṉar*
#trans= ‘ten **people** came’



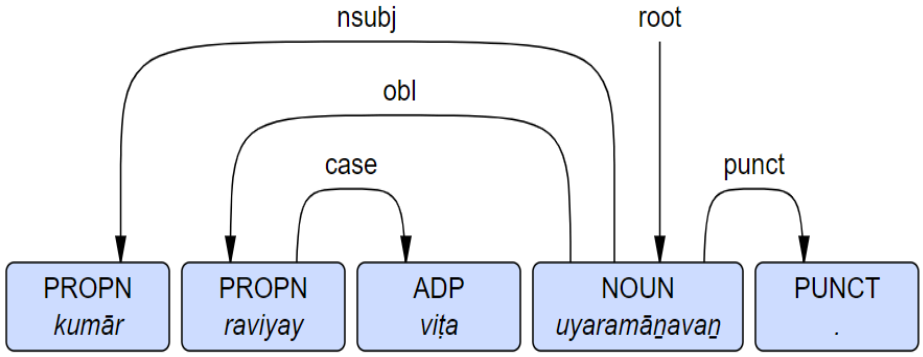
4.3.4.3 case: case marking

A syntactic word including prepositions, postpositions, and clitic case markers, which is used for case-marking are tagged with the dependency relation ‘case’.

Example (4.48)

#text= *kumār raviyay* ***viṭa*** *uyaramāṉavaṉ*

#trans= ‘Kumar is taller **than** Ravi’



4.4 Other tags

The following tags are not dependency relations. They are coordination tags and multi word expressions which are likely loose tags.

4.4.1 Coordination

‘conj’ and ‘cc’ are the two tags used to mark coordination in compound and complex sentences.

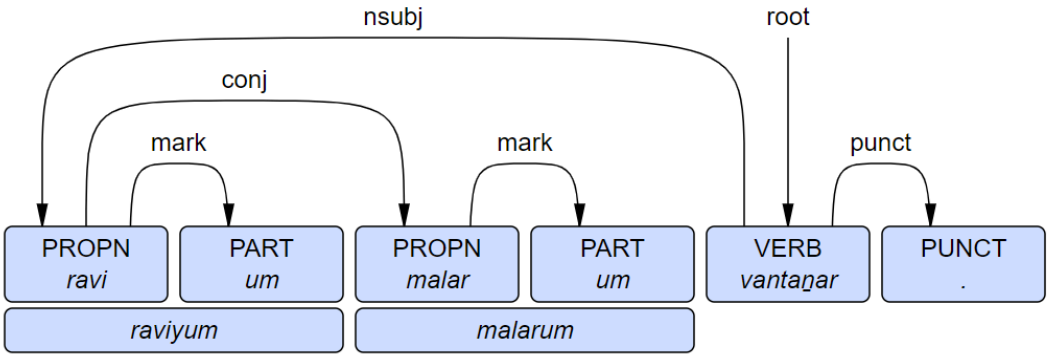
4.4.1.1 conj: conjunct

The dependency relation between two elements, which are connected by a coordinating conjunction, such as *and, or,* etc. are marked ‘conj’. In a list of coordinated items, the first one is treated as parent-head and the rest are marked ‘conj’ to that parent-head. It can be between verbs, nouns, clauses or sentences.

Example (4.49) noun-noun conj

#text= *raviyum* ***malarum*** *vantaṉar*

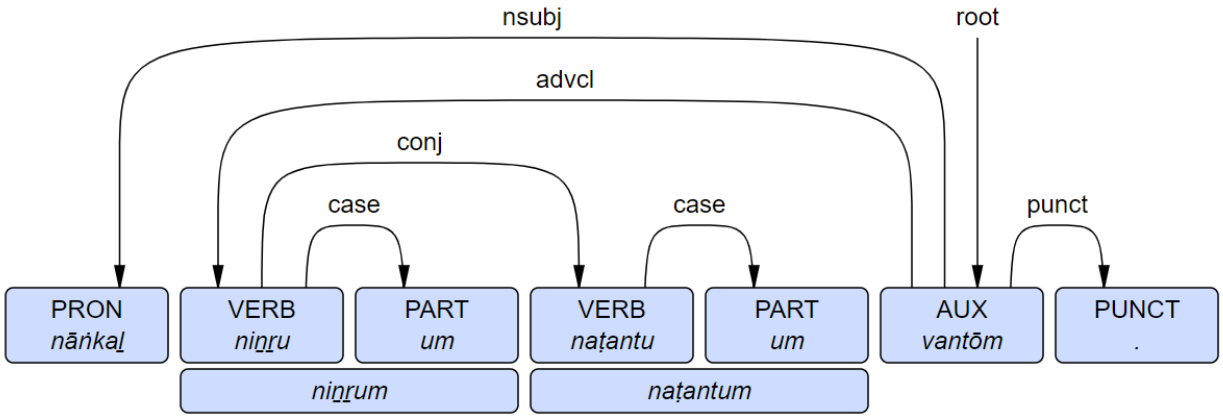
#trans= **‘Malar** and Ravi came’



Example (4.50) verb-verb conj

#text= *nāṅkaḷ niṉṟum* ***naṭantum*** *vantōm*

#trans= ‘We came by standing and **walking’**



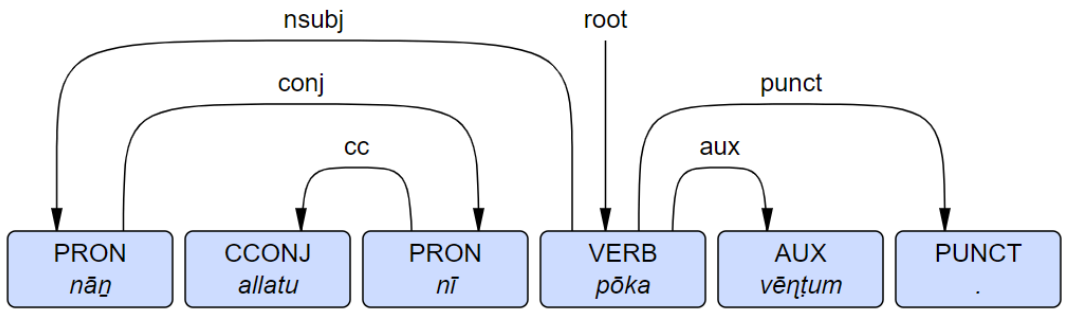
4.4.1.2 cc: coordinating conjunction

‘cc’ is the dependency relation between a conjunct and the preceding coordinating conjunction (conj).

Example (4.51)

#text= *nāṉ allatu nī pōka vēṇṭum*

#trans= ‘You or I should go’



4.4.2 Headless

Headless relations that are used to tag Multi-Word Expressions (MWEs) include ‘fixed’ and ‘flat’

4.4.2.1 fixed: fixed multiword expression

The relation ‘fixed’ is used for certain fixed grammaticized expressions that behave like function words or short adverbials.

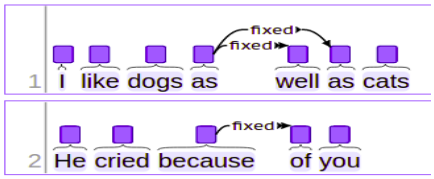


Figure 4.3 Sample English examples of ‘fixed’ dependency relation

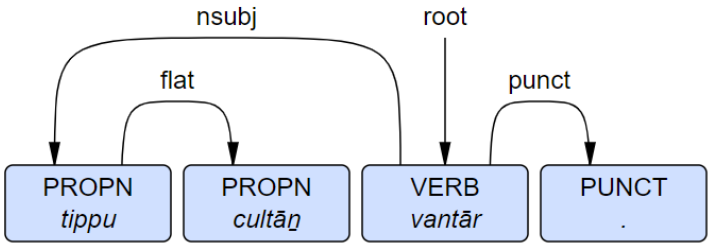
4.4.2.2 flat: flat multiword expression

The flat relation in UD is used for names like (*tippu cultāṉ*) and dates (*24 January*). It contrasts with fixed, which applies to completely fixed grammaticized MWEs.

Example (4.52)

#text= *tippu* ***cultāṉ*** *vantār*

#trans= ‘Tipu **Sultan** came’



4.4.3 Loose

Loose joining relations are used only if other relations are not possible. This category includes the tags, ‘list’ and ‘parataxis’.

4.4.3.1 list: list

The relation ‘list’ is used for chains/a long list of comparable items. It is a loose tag which is used white a set of items are listed. The first in the list is related to other items as ‘list’. This kind of sentence should be analyzed as a coordinate structure as much as possible. This tag is used only if ‘cc’ and ‘conj’ could not be used.

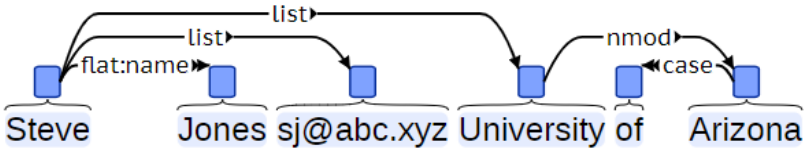


Figure 4.4: A sample example for ‘list’ extracted from www.universaldependencies.org

4.4.3.2 parataxis: parataxis

‘Parataxis’ is a dependency relation marked between a word that is from the main predicate of the sentence to a clause after {}/:/; found side by side without explicit coordination, subordination, or argument relation with the head word. An example from the Tamil dataset was not found.

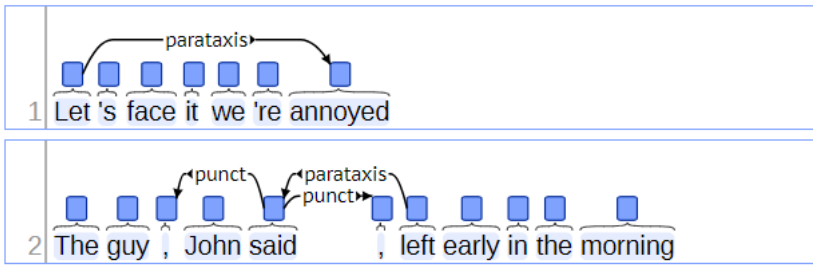


Figure 4.5: Sample English examples of 'prataxis' dependency relation extracted from www.universaldependencies.org

4.4.4 Special

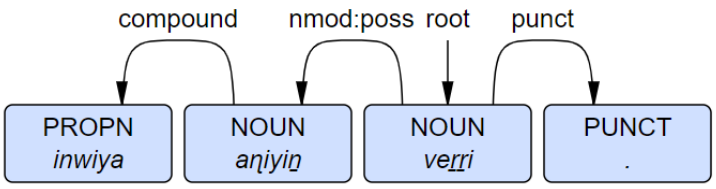
Special relations are given for ellipses, disfluencies and other orthographic errors. It also covers clausal heads, punctuations and compounding.

4.4.4.1 compound: compound

The compound relation is used for noun compounds (e.g., *phone book*) in general. This is also used for noun verb compounds which is a common occurrence in Tmail.

Example (4.54)

#text= *inwiya aɳiyiɳ verri*
#trans= 'Indian team's victory'



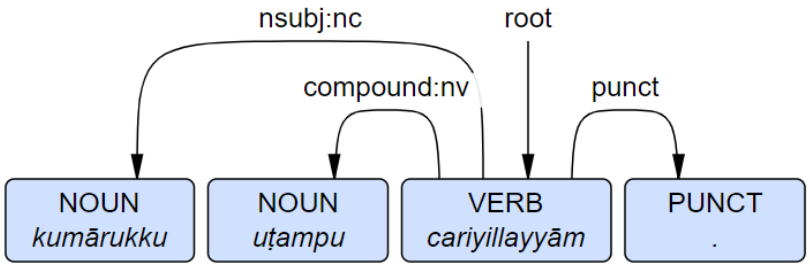
In Tamil, Noun-Verb (NV) compounds are commonly seen and thus, a separate language-specific tag ws developed.

(i) compound:nv

Example (4.55)

#text= *kumārukku uṭampu cariyillayyām*

#trans= ‘It seems that Kumar is sick’



4.4.4.2 orphan: orphan

The ‘orphan’ relation is used in predicate ellipsis where one of the core arguments has to be promoted to clausal head.

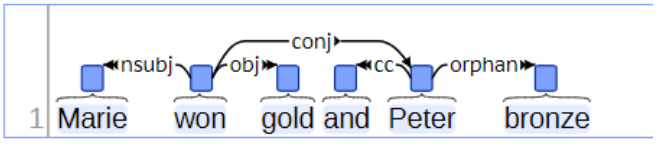


Figure 4.6: An example of ‘orphan’ extracted from www.universaldependencies.org

4.4.4.3 goeswith: goes with

The relation ‘goeswith’ relates the words that are segregated due to editing errors. Instead of being together as per grammatical tradition, those words are found separated. The later parts are connected to the head of that word with the relation ‘goeswith’.

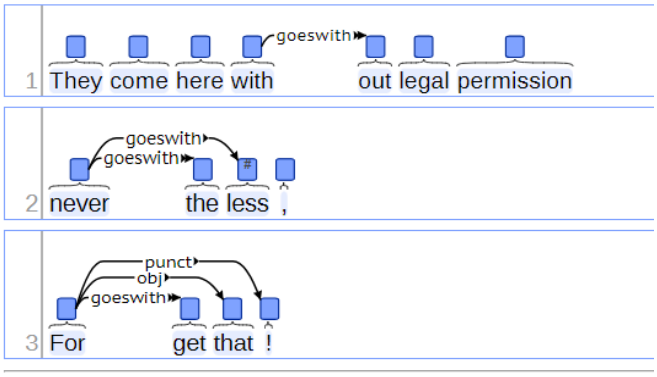


Figure 4.7: An example of ‘goeswith’ from www.universaldepedencies.org

4.4.4.3 reparandum: overridden disfluency

‘reparandum’ is used to indicate disfluencies overruled in a speech repair. The disfluency is dependent on the repair. It is not found in the annotated data.

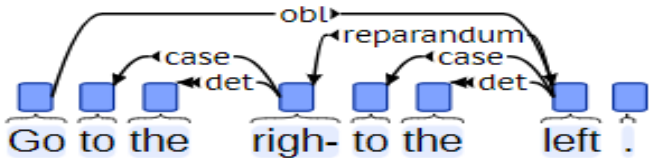


Figure 4.8: An example of ‘reprandum’ extracted from www.universaldependencies.org

4.4.5 Other

Other tags include punctuations, unspecified dependency relations and the root of the sentence.

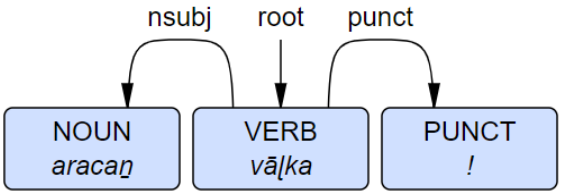
4.4.5.1 punct: punctuation

The relation ‘punct’ is used for any punctuation mark found in the text. The POS PUNCT are give the relation ‘punct’ and not SYM (symbols)

Example (4.56)

#text= *aracaṉ vā[ka!*

#trans= ‘Let the king live!’



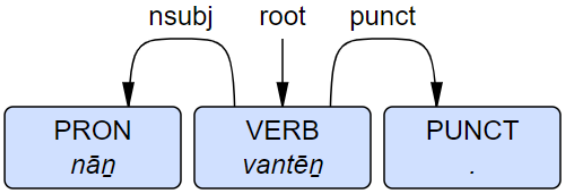
4.5.5.2 root: root

The ‘root’ grammatical relation points to the head of the sentence. At times, a fake node ROOT is used as the governor. In Tamil, verbs or nouns occur as ‘root’.

Example (4.57) Verb as root of the sentence

#text= *nāṉ vantēṉ*

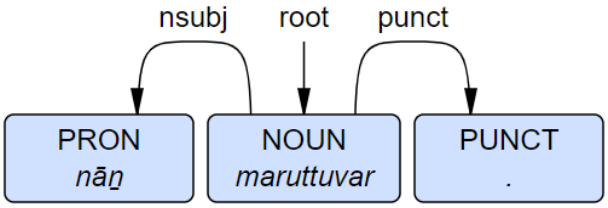
#trans= ‘I came’



Example (4.58) Noun as root of the sentence

#text= *nāṉ maruttuvar*

#trans= ‘I am a doctor’



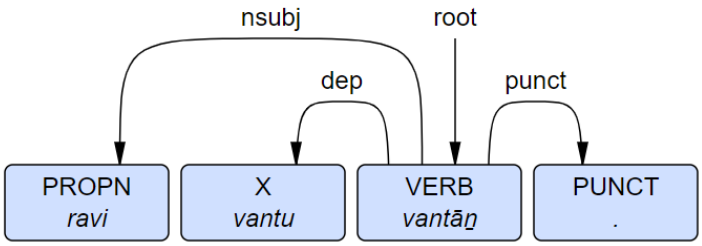
4.5.5.3 dep: unspecified dependency

The tag ‘dep’ is marked when a precise tag could not be tagged in the text. It happens generally in speech/ conversation texts. It is better to avoid the usage of this tag to the maximum.

Example (4.59)

#text= *ravi vantu vantāṉ*

#trans= ‘Ravi came’



Chapter 5

Building Tamil Syntactic Parser: Evaluation and Error Analysis

5.1 Introduction

Evaluating a parser is an important task in parsing. Parsers are evaluated using defined metrics. This chapter gives an outline on parsing evaluation methods and the evaluation score of the parser built on domain-specific data. This thesis also compares the data before and after fine tuning to domain-specific data. A statistical graph of tags used in each domain for both POS and syntactic relations are listed in this chapter. The errors in the result are analyzed and it's resolved.

5.2 Machine Learning models

Parsing models are programmed toolkits with a defined pipeline of multiple tools attached to it. Some of the recent parsing models include Stanza and Trankit.

5.2.1 Stanza

Stanza in dependency parsing is performed by the DepparseProcessor which gives syntactic dependency analysis. The requirements of the model include tokenization processor, MWT processor, POS processor and lemma processor. The syntactic head is determined by the model and the dependency relations are determined by the head and deprel relations. The memory usage is high when larger pre-trained models are used.

5.2.2 Trankit

Trankit is a python programmed toolkit which provides tainable pipeline for more than 100 languages in the field of NLP. It has over 90 pre-trained models for 56 languages. Trankit outperforms the other existing parsing models and does a better job in tokenization, morph and POS tagging, and syntactic tagging. It is used for more than 90 UD treebanks in UD V2.5. This thesis also uses Trankit, which is pre-trained by Tamil Syntactic Parser (developed at IIIT-H) as given in figure 5.1. The adapted model is fine-tuned with domain-specific data for better

accuracy. The model's memory usage and speed are much better than stanza even when larger pre-trained transformer models are used.

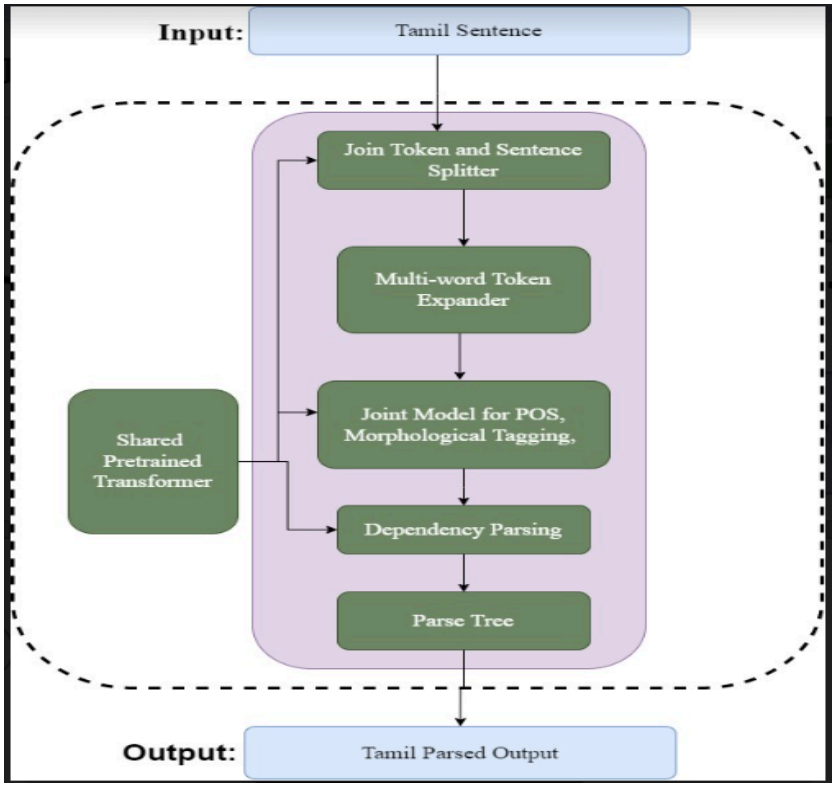


Figure 5.1: Architecture pipeline of Trankit parsing model used at IIIT-H

5.3 Statistics of the trained data

The TTR ratio, statistics of POS and syntactic tags for domain-wise corpus and overall corpus are presented in this section.

5.3.1 Type Token Ratio (TTR)

Type token Ratio is one of the requirements for selecting the corpus. It is the measure of vocabulary variation in a language's text. A good TTR is needed for developing a good parser. High TTR indicates the highness in lexical variations and low TTR indicates the opposite. The TTR ratio for each domain and overall TTR is listed in table 5.1 and 5.2 respectively. TTR is calculated as seen below:

$$TTR = \frac{\text{Total number of Types}}{\text{Total number of Tokens}} X100$$

Domain	Tokens	Types	TTR	Sentences
Tourism	11089	4113	37.09	1010
Agriculture	10002	4092	40.91	1001
Sports	10453	4326	41.39	1000
Social media	10809	4899	45.32	1007
Speech conversation	10342	5620	54.34	1003
Total based on domains	52,695	23,050	43.81	5,021

Table 5.1: Domain-wise TTR

Overall TTR	Tokens	Types	TTR	Sentences
Total corpus	52,695	23,050	43.74	5,021

Table 5.2: Total corpora's TTR

5.3.2 POS statistics of each domain

The frequency of occurrence of POS tags are compared between the domains and the table 5.3 is formulated below:

POS category	No. of occurrences				
Domains	Sports	Agriculture	Tourism	Social media	Speech conversation
NOUN	3526	2947	3048	2903	1995
DET	433	437	716	598	350
ADJ	540	571	632	754	531
ADV	648	395	543	747	629
ADP	139	277	390	580	477
VERB	1668	1663	1810	1864	1771
PUNCT	1230	1212	1124	1251	1321

PRON	342	187	732	612	782
PROPN	722	1193	426	878	1120
AUX	414	459	619	423	502
CCONJ	231	138	415	188	410
NUM	105	114	301	427	119
SCONJ	111	103	107	119	176
PART	236	270	111	254	108
SYM	8	36	115	211	43
INTJ	-	-	-	-	-
X	-	-	-	-	8

Table 5.3: POS tags used in the data for all the 5 domains

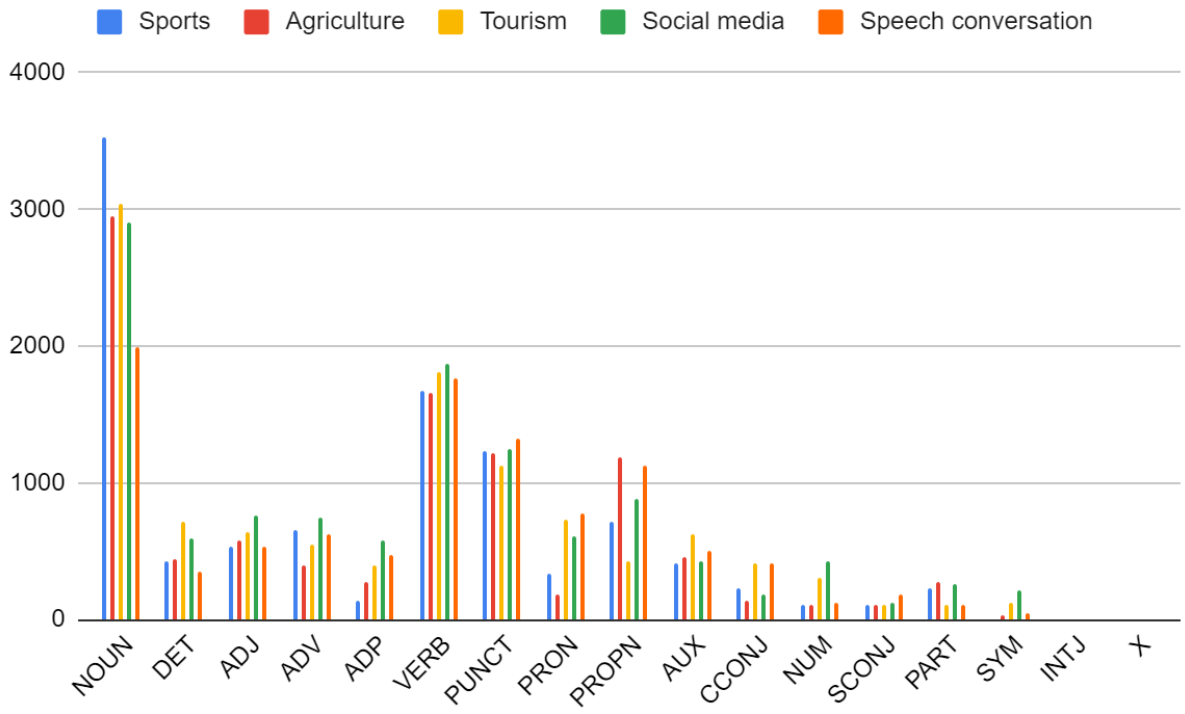


Figure 5.2: Bar chart representation of POS data

The tag NOUN touches the highest number in the Sports domain and the least in speech conversation. Pronouns are used instead of nouns in speech conversation. Proper nouns (PROPN) are found more in Agriculture domain as the biological terms are more in number.n spite of having places and people's names in tourism domain, the occurrences are low when

compared to other domains. NUM and SYM are the highest in social media text due to the smileys and hashtags.

5.3.3 Statistics of dependency relations

Dependency Relation	Number of occurrences				
	Sports	Agriculture	Tourism	Social media	Speech conversation
acl	177	322	248	191	189
advcl	389	211	148	340	332
advmod	648	395	543	747	629
amod	540	571	632	754	531
appos	8	5	10	-	4
aux	399	378	455	330	467
aux:pass	15	81	164	93	35
case	139	277	390	580	477
cc	236	233	122	50	389
ccomp	344	42	122	144	197
compound	100	241	357	150	133
conj	241	254	141	56	78
cop	154	279	226	103	17
csubj	23	14	56	9	11
det	433	437	716	598	350
discourse	-	-	-	-	-
flat	17	46	94	78	35
iobj	316	211	264	156	48
mark	236	270	111	254	108
nmod	414	277	127	542	32
nmod:poss	347	191	254	342	80
nsubj	905	741	845	607	703

nsubj:pass	75	76	37	-	4
nummod	105	114	301	427	119
obj	538	200	318	345	376
obl	427	32	62	37	145
obl:abl	-	17	100	121	57
obl:agent	60	62	78	76	12
obl:com	12	-	12	12	46
obl:inst	64	50	78	111	91
obl:loc	309	201	290	358	127
obl:number	-	5	-	8	14
parataxis	-	-	-	-	-
punct	1230	1212	1124	1251	1321
root	1000	1001	1010	1007	1003
xcomp	170	187	226	121	38

Table 5.4:statistics of dependency tags used in the data

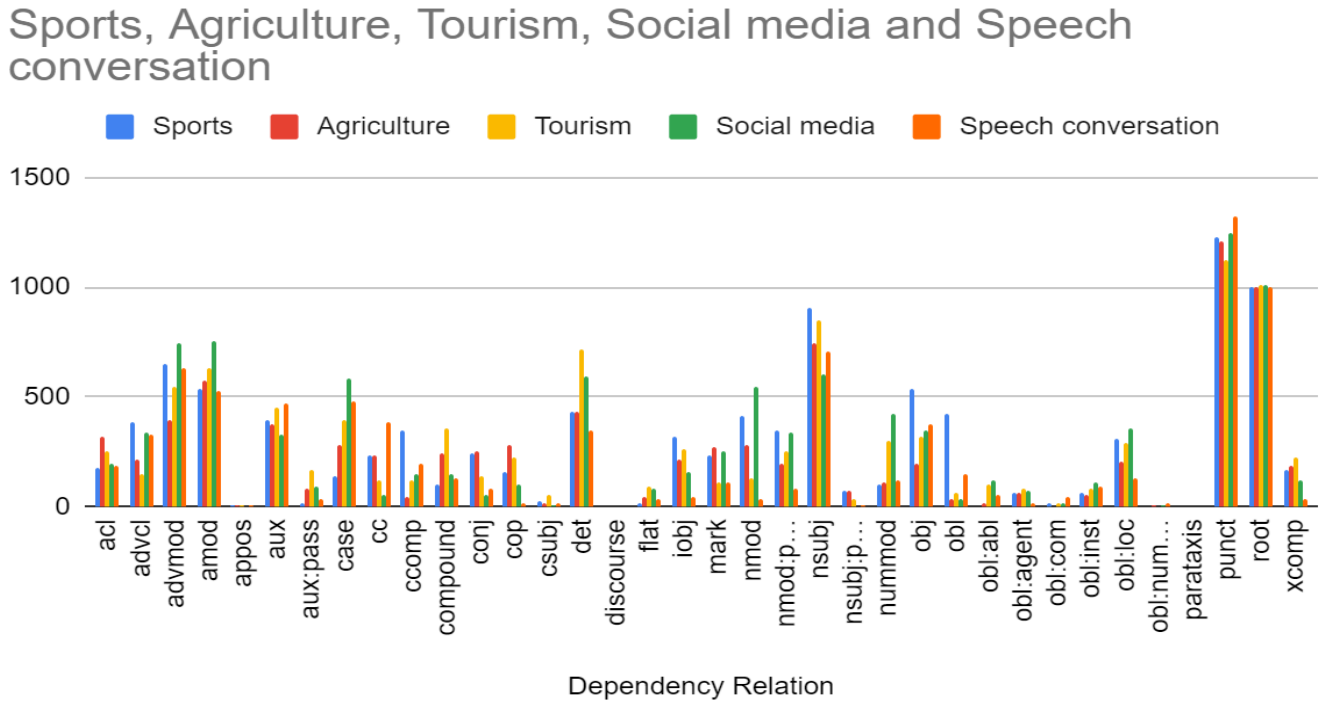


Figure 5.3: bar chart representation of dependency relations used in the domain-specific data

The figure 5.3 draws a comparison on different kinds of constructions used in different domains. While relative clauses are found more in Sports data, adnominal clauses are found more in agriculture data. Relative clauses, clausal complements and Passive constructions are seen more in tourism data. Infinitive and copula constructions are almost seen in all the data. Adverbial modifiers and adjectival modifiers are seen more in social media text.

5.4 Evaluation metrics: Attachment Scores

The parsers that are developed need to be evaluated. Attachment scores is one of the evaluation metrics used to evaluate the parser's accuracy (Nivre, 2009). It includes Labelled Attachment Scores (LAS) and Unlabelled Attachment Scores (UAS). UAS indicates the percentage of words that are assigned to correct heads and LAS indicates the percentage of words that are assigned to correct heads with correct dependency label.

5.5 Results

The process of fine-tuning the Tamil data with domain-specific corpus has resulted in a significant increase in accuracy. The accuracy after fine-tuning is increased by LAS 2.7% and UAS 1.8% as shown in the tables below:

Model	**Tokens**	**Lemma**	**UPOS**	**UFeats**	**UAS**	**LAS**
Trankit	98.02	88.85	86.18	87.19	72.34	67.66
Stanza	99.58	85.14	82.60	81.89	61.23	55.76
Tamil Parser by IIIT-H	Rule based (99.23)	91.10	94.47	95.19	87.4	79.60
Parser adapter for Domain-specific data	**99.23**	**91.12**	**94.47**	**95.2**	**89.2**	**82.3**

Table 5.5: The comparison of previously trained models with parser adapted for domain-specific data

5.6 Error analysis

This section discusses the areas in which domain-specific parsers fail to give out appropriate results. This section has three sections:

5.6.1 Pre-processing errors

Pre-processing errors like tokenisation errors in sentences and words, wrong clitics split, wrong splits in auxiliaries or passive auxiliaries, multi-token word errors are the reasons for significant reduction in accuracy rates. As the split goes wrong, the morph, POS, dependency relations and overall sentence becomes an error.

Example (5.1) *55vitaykaḷ* '55seeds'

Here, 55 and *vitaykaḷ* needs a split. Such split errors are the reason for unknown words in morph. Morphological errors like unknown words form the major component of failure of the morph module. Ex 5.1 is an example of morph error too as such a token is not listed in the morph dictionary.

Example (5.2) *pa.ja.ka* 'BJP'- wrong split of abbreviations

In Ex:5.2, after each full stop, the letters are split into different sentences. Wrong sentence boundary identification is the cause for such issues.

5.6.2 POS errors

POS errors like wrong identification of POS was a common issue. This wrong identification was a reason for morph errors too.

(i) PROPN Vs. NOUN

Nouns and Proper Nouns identification needs a huge set of language's vocabulary to be learnt by the machine. Updating vocabulary in the dictionary list is an essential step to avoid such issues. Ambiguous words are another reason for wrong identification, which requires semantics to unfold the ambiguity.

Example (5.3) *puṉita nīr kēṉi* ‘Punitha neer well’
Here, *puṉita nīr* is not sacred water as it literally means. ‘*puṉita nīr’*- both the words are marked PROPN and ‘*kēṉi’* NOUN. *puṉita nīr* is the name of a well in the tourist spot. Such errors lead to a significant reduction in accuracy levels especially in domain-specific data.

(ii) NOUN Vs. VERB

Nominalised verbs are supposed to be marked as NOUN in POS. But, the machine marks it VERB, leading to error.

Example (5.4) *ravi* ***ceyvatu*** *tavaṟu* 'What Ravi **did** was wrong’

The word *ceyvatu* is marked VERB instead of NOUN. Such errors are very common across the domains.

(iii) VERB Vs. AUX

Copulas and auxiliaries are marked as VERB. Some of the auxiliaries can act as verbs in other contexts. Semantic knowledge is required to resolve those ambiguities.

Example (5.5) *ceytu muṭittēṉ* ‘(I) completed’
Example (5.6) *muṭittu viṭṭēṉ* ‘(I) finished’

muṭi in Ex:5.5 is AUX and in Ex:5.6 is VERB. Such contextual meaning is required to disambiguate this issue.

(iv) ADV Vs. ADJ

Adverbs and Adjectives result in error when it comes to intensifiers.

Example (5.7) *mika mika nallatu* ‘very very good’

mika is an intensifier, which is marked ADJ and ADV at random occurrences. The decision taken is when it occurs before a noun, it's an ADJ and when it occurs before an adjective or a verb or another adverb, it is considered as ADV.

(v) ADP Vs. VERB

Adpositions are marked as VERB for certain adpositions due to lexical ambiguity.

Example (5.8) *kappalay* ***curri*** *vantār* '(He) went around the ship'

Such words can also occur as verbs meaning 'wandered'. Machine needs semantic context to get correct results.

(vi) SCONJ Vs. ADV

The words that connect the previous and the current sentence acts as a discourse marker and are marked SCONJ. At times, it is confused with adverbs which occur at sentence initial position.

Example (5.9) ***āṉāl****, avaṉ varavillay* 'But, he didn't come'

āṉāl is marked SCONJ and it is tagged 'advmod' to the root of the sentence.

5.6.3 Dependency relation errors

Dependency tags that are used in unique constructions are not learnt well by machine algorithm as their occurrences are very small in number. Also, some tags are similar in features. Some of those are found as errors.

(i) Copula constructions

A good knowledge on Tamil syntax and semantics is necessary to identify copulas and their roles in linking subjects and predicates. Tamil copula *ākum* is connected to the predicate noun as copula with the POS as AUX.

Example (5.10) *itu ōr paruvakkāla nīrvīḻcciy**ākum***

'This **is** a seasonal waterfall'

Decision:

(i) The predicate noun is the "root" of the sentence

(ii) *ākum* as copula, connected to the predicate

(ii) nmod vs. compound

Nominal modifiers and compounds perform similar functions. Identifying compound words and understanding their compositional meanings necessitates linguistic expertise and context awareness. A good grammatical analysis is needed to differentiate the two. PROPN, NOUN, PRON in POS are marked with nmod/compound.

Example (5.11) ***inwiya*** *vakay arici* **'Indian** variety of rice'

Here, *inwiya* is marked with 'compound'

Example (5.12) *itu **malariṉ** puttakam* 'This is Malar's book'

Here, *malariṉ* is 'nmod'

Decision:

(i) Nouns with or without possessive marker expressing possession is marked 'nmod'

(ii) Nouns that is realized as two different words but morpho syntactically one word or that gives more specific information about a common noun is marked 'compound'

(iii) Multi-token words

Careful attention needs to be given to separate the multi-token words in morphology and word boundaries of Tamil.

Example (5.13) *aḻavayyākum* = ***aḻavay+ ākum*** 'Measurement unit'

Decision:

Syntactically two different word forms which are written together are split.

(iv) Conjunction : Head initial Approach

Analyzing the positioning of heads and constituents, it demands a sound knowledge of Tamil syntactic structures.

Example (5.14) *avaṉ**um** avaḷ**um** vantārkaḷ*

Decision:
avaṉum avaḷum are conjoined (‘conj’) with conjunction marker *-um*

(v) xcomp vs advcl

Differentiating between types of clausal complements and adverbial clause modifiers demands precise syntactic analysis in Tamil sentences.

Example (5.15) *ravi malaray **talayvi** ākkiṉāṉ* ‘Ravi made Malar a **leader’**

Example (5.16) *ravi **vantu** ceṉṟāṉ* ‘Ravi **came** and went’

In Ex:5.6, leader has a special quality of taking the ‘xcomp’ tag and in Ex:5.7, *vantu* is a verbal participle which is marked ‘advcl’.

Decision:

Verbal participles are tagged ‘advcl’ and the subordinate clause which shares the subject with the matrix clause are marked ‘xcomp’.

(vi) mark vs case

Distinguishing between markers and cases in Tamil requires detailed knowledge of the language's grammatical features.

Example (5.17) *tampi akkā varukiṟār **eṉṟu** coṉṉāṉ* ‘Brother said **that** the sister is coming’- **mark**
Example (5.18) *kumār raviyay **viṭa** uyaramāṉavaṉ* ‘Kumar is taller **than** Ravi’ - **case**

Decision:
Here, ‘case’ is marked for adpositions and clitics other than *‘um’* and ‘mark' is marked for complementizers or other elements which helps in subordinating the clause and for clitic ‘*-um’*.

(vii) cc and conj vs list

Recognizing coordinating conjunctions and their usage within compounds poses challenges, especially in complex sentence structures. The tag list and cc/conj are very tricky to tag.

Decision:

The tag ‘list’ is a loose tag and it is to be avoided to the maximum. The tags ‘cc’ and ‘conj’ will be used for sentences which have a list of entities that are separated by comma.

Such errors are looked into and they are re-analysed to obtain a better accuracy. Also, our future includes accommodating such issues .

Chapter 6

Conclusion

This research work has explored various grammatical frameworks, parsing techniques, and tagsets for developing treebank and finally, has chosen the dependency grammar and universal dependency tagset as the appropriate framework to develop an effective parser for Tamil. Domain adaptation of existing parser is an important milestone in Indian languages as it has not been worked upon in the other Indian languages. In this chapter, some concluding remarks, major contributions, significance of this research, and future works are looked into.

As a part of the introduction chapter, an analysis of various grammar formalisms, parsing techniques, annotation schema, kinds of parsers, and theoretical and computational frameworks are presented. A brief introduction about the 'domain' and some examples of unique constructions in each domain is illustrated. In addition, an introduction to Tamil with its syntactic features is presented. Also, a brief methodology is presented.

The second chapter discusses the literature review, where the papers/ theses/ articles/ books/ news articles/ websites that are published on parser, parsing techniques, works that are majorly done on Universal dependency parsing in global languages, Indian languages, Dravidian languages with special reference to Tamil are listed.

The third chapter discusses the morphological and POS tagsets, where it talks about the guidelines that are used to tag the morphology and the parts of speech in the treebank. A treebank of 100k tokens was developed initially using this guideline for Tamil Syntactic Parser at IIIT, Hyderabad. The same model was adapted for domain-specific data (1k from each domain including tourism, agriculture, sports, social media and speech conversation). POS, and morphological tags were tagged using Universal Dependency tagsets. Certain tags were not used based on the necessity of language-specific features. New feature sets like "Gender=Fem,Masc" were introduced in morphology as it was required to satisfy the morphology of Tamil.

The fourth chapter is the most important chapter which has the syntactic tags. Developing syntactic tags set for Tamil with language-specific features was a challenging task. The tags that are developed in Universal Dependencies are based on other language patterns. The advantage of this tagset is that it allows to develop language-specific guidelines according to the language's necessity. So, the tags that are already explained are re-analysed and it was applied to Tamil data with certain modifications. As the morphology and syntax demands, the tags are modified and language-specific tags like "obl:tmod" are introduced.

The fifth chapter discusses the evaluation of the domain-specific data after adapting the existing parser model. It has the record of the accuracy of the parser. This chapter in short discusses the error analysis done after evaluation.

6.1 Advantages and disadvantages of domain-specific data

Advantages:

- This method of adapting a domain-based treebank is a good platform for developing multi-domain treebank
- Domain-based treebank gives better accuracies when tested for each domain with their respective domain-specific test data
- Technical terms are addressed in annotation which reduces the usage of Named Entity Recognition (NER) modules
- After domain adapting the existing parser, it is witnessed that accuracies have increased.

Disadvantages:

- Domain- specific treebank gives a worse accuracy when other domain texts are used for testing
- Multiple domains are required to give a better accuracy when across the domain data is used

6.2 Major contributions

- This thesis has done a contribution of annotating POS and syntactic tags to the MWTT treebank in Universal Dependencies.

- Thesis also contributes to the recently released Tamil Syntactic parser, developed by IIIT Hyderabad. The initial phase of work, corpus cleaning and POS tagging was done.
- A brief comparison of the UD and AnnCorra tagset is provided in addition to analyzing the pros and cons of each schema.
- This study has enhanced the existing UD Tamil by adding a few language-specific tags
- Domain adaptation of the existing IIIT parser contributes to the development of domain-specific treebanks.
- An in-detail framework for marking dependency relations for Tamil including illustrations and exceptions are presented.
- The domain-specific parser developed as part of this study can also be adapted to other Dravidian languages with minimal effort

6.3 Challenges

- Since the accuracy of the parser relies on the each of the pre-processing tools used earlier, a small error led to a big accuracy difference
- As the language vocabulary is evolving, the database needs to be updated too, which requires the changing of rules and introducing new paradigms in the pre-processing tools.
- Filtering the right morphological analysis was a challenging task and certain contexts demanded two different analyses due to ambiguities. Dealing with such ambiguities were really challenging.
- Failure in each pre-processing led to a decrease in recall and precision
- Improvement of pre-processing tools was crucial in improving the performance of the parser

6.4 Future Work

- More domains can be explored
- This is an initial work on the Universal Dependency framework for Tamil. The future work can extend the number of sentences in other domains to improve the accuracy
- Inclusion of dialectal data which would deal with ellipses is a crucial futuristic study
- Introducing more language-specific tags can enrich the existing data

- Fine-grained data will improve the accuracy. Morphological features can be improved in the future with fine-grained details
- The same tags can be extended to other Dravidian languages as well

References

Amba Kulkarni, Sanal Vikram, and Sriram K. 2019. Dependency Parser for Sanskrit Verses. In Proceedings of the 6th International Sanskrit Computational Linguistics Symposium, pages 14–27, IIT Kharagpur, India. Association for Computational Linguistics.

Ambati, B. R., Gadde, P., & Jindal, K. 2009. Experiments in Indian Language Dependency Parsing. In the Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing, pp. 32-37.

Ambati, B.R., 2016. Transition-based combinatory categorial grammar parsing for English and Hindi. https://era.ed.ac.uk/handle/1842/20404. Retrieved on 15th June,2022.

Ambati, B.R., Deoskar, T. and Steedman, M., 2018. Hindi CCGbank: A CCG treebank from the Hindi dependency treebank. Language Resources and Evaluation, 52, pp.67-100.

Amita, A. J. 2015. An Annotation Scheme for English Language Using Paninian Framework. IJISET, Vol. 2, pp. 616-619.

Atul Kr. Ojha and Daniel Zeman. 2020. Universal Dependency Treebanks for Low-Resource Indian Languages: The Case of Bhojpuri. In Proceedings of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation, pages 33–38, Marseille, France. European Language Resources Association (ELRA).

Bahrani, M., Sameti, H., and Manshadi, M. H. 2011. A Computational Grammar for Persian Based on GPSG. Retrieved on 21st November, 2017. https://link.springer.com/article/10.1007/s10579-011-9144-1

Bhadriraju Krishnamurti. 2003. The Dravidian Languages. Cambridge University Press.

Bharati, A., Sangal, R. 1993. Parsing Free Word Order Languages in the Paninian Framework. In: Proceedings of the 31st Annual Meeting on Association for Computational Linguistics, pp. 105–111.

Bharati, A., Sangal, R., Chaitanya, V., Kulkarni, A., Sharma, D. M., and Ramakrishnamacharyulu, K. V. 2002. AnnCorra: Building Tree-banks in Indian Languages. In Proceedings of the 3rd Workshop on Asian language Resources and International Standardization: Association for Computational Linguistics. Vol 12. pp. 1-8.

Bharati, A., Sharma, D. M., Husain, S., Bai, L., Begam, R., and Sangal, R. 2009. Anncorra: Treebanks for Indian Languages, Guidelines for Annotating Hindi Treebank. Retrieved on 1st December, 2016. http://aclweb.org/anthology/W17-63

Bharati, A., Vineet Chaitanya and Sangal, R. 1995. Natural Language Processing: A Paninian Perspective. New Delhi: Prentice-Hall of India.

Bhat, I.A., Bhat, R.A., Shrivastava, M. and Sharma, D.M., 2018. Universal Dependency parsing for Hindi-English code-switching. arXiv preprint arXiv:1804.05868.

Bhat, R. A., Bhat, I. A., and Sharma, D. M. 2017. Improving Transition-Based Dependency Parsing of Hindi and Urdu by Modeling Syntactically Relevant Phenomena. In ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), Vol 16, No. 3, pp 17.

Bikel, D. M., and Chiang, D. 2000. Two statistical parsing models applied to the Chinese Treebank. In ACL, Vol 12. pp. 1-6.

Bresnan, J. 1982. Control and complementation. In Linguistic inquiry. Vol. 13. No. 3. pp. 343-434.

Briscoe, E, Carroll, J and Watson, R. 2006. The Robust Accurate Statistical Parsing (RASP). Retrieved on 20th February, 2018. http://sro.sussex.ac.uk/25679/

Bunt, H., Carroll, J., Satta, G. (eds) New Developments in Parsing Technology. 2005. Text, Speech and Language Technology, vol 23. Springer, Dordrecht. https://doi.org/10.1007/1-4020-2295-6_1

Cha, J., Lee, G. & Lee, J. 2002. Korean Combinatory Categorial Grammar and Statistical Parsing. Computers and the Humanities 36, 431–453. https://doi.org/10.1023/A:1020260012525

Charniak, E. 2000. A maximum-entropy-inspired parser. In ACL, pp. 132-139.

Chomsky, N., 1959. A note on phrase structure grammars. Information and control, 2(4), pp.393-395.

Clark, Alexander, Chris Fox, and Shalom Lappin. 2013. The Handbook of Computational Linguistics and Natural Language Processing. John Wiley & Sons. 6.

Collins, M. 1997. Three Generative, Lexicalised Models for Statistical Parsing. In Proceedings of the 8th Conference on European Chapter of the ACL, pp. 16-23.

Collins, M., Ramshaw, L., Hajič, J., and Tillmann, C. 1999. A statistical parser for Czech. In ACL, pp. 505-512.

Dan Klein and Christopher D. Manning. 2003a. Accurate Unlexicalized Parsing. In Proceedings of the 41st ACL, pp. 423-430.

Dan Klein and Christopher D. Manning. 2003b. Fast Exact Inference with a Factored Model for Natural Language Parsing. In Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge: MIT Press, pp. 3-10.

Danqi Chen and Christopher D. Manning 2014. A Fast and Accurate Dependency Parser using Neural Networks. In EMNLP, pp. 740-750.

De Marneffe, M. C., and Manning, C. D. 2008. Stanford Typed Dependencies Manual. California: Stanford University. Technical report, pp. 338-345.

de Marneffe, M.C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J. and Manning, C.D., 2016. Universal Dependencies: A cross-linguistic typology. https://nlp.stanford.edu/~manning/papers/USD_LREC14_UD_revision.pdf. Retrieved on jan, 2023.

Debusmann, R., 2000. An introduction to dependency grammar. Hausarbeit fur das Hauptseminar Dependenzgrammatik SoSe, 99(1), p.16.

Dhivya, R. 2011. Dependency Parser for Tamil Using Machine Learning Approach. Retrieved on 15th November, 2016. http://nlp.amrita.edu:8080/project/mhrd/ms/Tamil/DependencyParser_Dhivya_CEN09009.pdf

Dubey, A., and Keller, F. 2003. Probabilistic Parsing for German Using Sister-Head Dependencies. In Proceedings of the 41st ACL. Vol 1, pp. 96-103.

Foth, K. A., and Menzel, W. 2006. Hybrid parsing: Using Probabilistic Models as Predictors for a Symbolic Parser. In ACL, pp. 321-328.

Garapati, U. R., Koppaka, R., and Addanki, S. 2012. Dative Case in Telugu: A Parsing Perspective. In Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages, pp. 123-132.

Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. A. 1985. Generalized Phrase Structure Grammar. Cambridge: Harvard University Press.

Gervasi, V., 2001. The Cico domain-based parser. Retrieved on Dec 12, 2023

Goyal, P., Mital, M.R., Mukerjee, A., Raina, A.M., Sharma, D., Shukla, P. and Vikram, K., 2003. A bilingual parser for Hindi, English and code-switching structures. In 10th Conference of The European Chapter. pp. 15.

Güngördü, Z. and Oflazer, K., 1995. Parsing Turkish using the lexical functional grammar formalism. Machine Translation, 10, pp.293-319.

H. Yamada and Y. Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In Proceedings of the 8th International Workshop on Parsing Technologies (IWPT), pp. 195–206.

Hockenmaier, J., and Steedman, M. 2002. Generative Models For Statistical Parsing with Combinatory Categorial Grammar. In Proceedings of the 40th ACL, pp. 335-342.

Husain, S. 2009. Dependency Parsers for Indian Languages. In Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing.

India, POMPI. 2011. Census of India 2011, paper1 of 2018, language. New Delhi: Office of the Registrar general and Census Commissioner. 4.

Jurafsky, D. 2000. Speech and Language Processing. London: Pearson.

Khan, Naira & Khan, Mumit. 2006. Developing a Computational Grammar for Bengali Using the HPSG Formalism. https://www.researchgate.net/publication/47523530_Developing_a_Computational_Grammar_for_Bengali_Using_the_HPSG_Formalism/citation/download. Retrieved on 16th June, 2022.

Klein, D., and Manning, C. D. 2003. Accurate Unlexicalized Parsing. In Proceedings of the 41st ACL. Vol 1, pp. 423-430.

Krishnamurti, Bhadriraju. 2003. The Dravidian Languages. Cambridge Language Surveys (1 ed.). Cambridge: Cambridge University Press. ISBN 978-0-521-77111-5.

Krivanek, J., and Meurers, D. 2011. Comparing Rule-Based and Data-Driven Dependency Parsing of Learner Language. Retrieved on 1st May, 2018. http://www.depling.org/proceedingsDepling2011/papers/krivanekMeurers.pdf

Krivanek, J., and Meurers, D. 2013. Comparing Rule-Based And Data-Driven Dependency Parsing of Learner Language. In Computational Dependency Theory. pp. 258-207.

Kübler, S., Ryan McDonald, Joakim Nivre, Graeme Hirst. 2009. Dependency Parsing. In Synthesis Lectures on Human Language Technologies. California: Morgan and Claypool Publishers. Vol 1. No.1.

Kulkarni, A., 2021. Sanskrit Parsing: Based on the Theories of Śābdabodha. DK Printworld (P) Ltd.

Kumari, B., and Rao, R. R. 2015. Improving Telugu Dependency Parsing Using Combinatory Categorial Grammar Supertags. In ACM Transactions on Asian and Low-Resource Language Information Processing, Vol 14. No.1, pp. 3.

Kumari, B.V.S. and Rao, R.R., 2015. Improving Telugu dependency parsing using combinatory categorial grammar supertags. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 14(1), pp.1-10.

Levine, R. D., and Meurers, W. D. 2006. Head-Driven Phrase Structure Grammar. In Encyclopedia of Language and Linguistics. Vol. 5, pp. 237-52.

Liu, H., and Huang, W. 2006. A Chinese Dependency Syntax for Treebanking. In Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation, pp. 126-133.

Makwana, M.T. and D.C.Vegda. 2015. Survey: Natural Language Parsing for Indian Languages. Retrieved on 12th January, 2018. https://arxiv.org/ftp/arxiv/papers/1501/1501.07005.pdf

Mannem, P. 2009. Bidirectional Dependency Parser for Hindi, Telugu and Bangla. Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, Daniel Zeman; Universal Dependencies. 2021. Computational Linguistics. 47 (2): 255–308. doi: https://doi.org/10.1162/coli_a_00402

Mates, Benson. 1961. Stoic Logic. Berkeley: University of California Press.

McDonald, R., Crammer, K., and Pereira, F. 2005. Online Large-Margin Training Of Dependency Parsers. In Proceedings of the 43rd ACL, pp. 91-98.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., ... and Bedini, C. 2013. Universal Dependency Annotation for Multilingual Parsing. In Proceedings of the 51st ACL. Vol. 2, pp. 92-97.

Menon, V.K., Rajendran, S., Kumar, M.A. and Soman, K.P., 2016. A new TAG Formalism for Tamil and Parser Analytics. arXiv preprint arXiv:1604.01235.

Mojgan Seraji, Filip Ginter, and Joakim Nivre. 2016. Universal Dependencies for Persian. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 2361–2365, Portorož, Slovenia. European Language Resources Association (ELRA).

Müller, S., Abeillé, A., Borsley, R.D. and Koenig, J.P., 2021. Head-Driven Phrase Structure Grammar: The handbook (Volume 9). Language Science Press.

Nagaraju, G., Mangathayaru, N., and Rani, B. P. 2016. Dependency Parser for Telugu Language. In ACM, Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, pp. 138.

Neidle, C. 1994. Lexical-Functional Grammar (LFG): In RE Asher, editor, Encyclopedia of Language and Linguistics. Oxford: Pergamon Press, pp. 9-25.

Nivre, J. 2020. Multilingual Dependency Parsing from Universal Dependencies to Sesame Street. In: Sojka, P., Kopeček, I., Pala, K., Horák, A. (eds) Text, Speech, and Dialogue. TSD 2020. Lecture Notes in Computer Science(), vol 12284. Springer, Cham. https://doi.org/10.1007/978-3-030-58323-1_2

Nivre, J. 2006. Inductive Dependency Parsing. In Text, Speech and Language Technology. Netherlands: Springer. Vol 34.

Nivre, J. 2009. Parsing Indian Languages with Maltparser. In the Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing, pp. 12-18.

Nivre, J., 2020. Multilingual dependency parsing from universal dependencies to sesame street. In Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8–11, 2020, Proceedings 23. pp. 11-29. Springer International Publishing.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. and Marsi, E. 2007. MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. Natural Language Engineering, Vol. 13. No. 2, pp. 95-135.

Nivre, Joakim. 2009. Parsing indian languages with maltparser. Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing.pp 12–18. 21, 165

Ojha, A.K. and Zeman, D., 2020, May. Universal Dependency treebanks for low-resource Indian languages: The case of Bhojpuri. In Proceedings of the WILDRE5–5th workshop on Indian language data: resources and evaluation. pp. 33-38.

Osborne, T.J., 2018. Tests for constituents: What they really reveal about the nature of syntactic structure. Language under discussion, 5(1), pp.1-41.

P. Jain, R. Bhavsar, A. Kumar, B. V. Pawar, H. Darbari and V. C. Bhavsar, 2018. Tree Adjoining Grammar Based Parser for a Hindi Text-to-Scene Conversion System, 2018. 3rd International Conference for Convergence in Technology (I2CT), Pune, India, pp. 1-7, doi: 10.1109/I2CT.2018.8529491.

P.Sangeetha. 2022. A Rule-based Dependency Parser for Telugu. PhD Thesis

Parida, S., Sahoo, K., Ojha, A.K., Sahoo, S., Dash, S.R. and Dash, B., 2022. Universal Dependency Treebank for Odia Language. arXiv preprint arXiv:2205.11976.

Phillips, J. D. 1992. A Computational Representation for Generalised Phrase-Structure Grammars. Linguistics and Philosophy. Vol.15. No.3, pp. 255-287.

Phillips, J.D. 1992. A computational representation for generalised phrase-structure grammars. Linguist Philos 15, 255–287. https://doi.org/10.1007/BF00627679

Pollard, C., and Sag, I. A. 1994. Head-Driven Phrase Structure Grammar. US: University of Chicago Press.

Prague Tagset. Retrieved on 1st June, 2018. http://ufal.mff.cuni.cz

Proudian, D., and Pollard, C. 1985. Parsing Head-Driven Phrase Structure Grammar. In Proceedings of the 23rd ACL. pp. 167-171.

Ramasamy, L., and Žabokrtský, Z. 2011. Tamil Dependency Parsing: Results Using Rule Based And Corpus Based Approaches. In International Conference on Intelligent Text Processing and Computational Linguistics. Berlin: Springer, pp. 82-95.

Ramasamy, L., and Žabokrtský, Z. 2012. Prague Dependency Style Treebank for Tamil. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012).

Rangra, R. 2015. Basic Parsing techniques in natural Language Processing. International Journal of Advances in Computer Science and Technology, 4(3).

Rayson, P., Archer, D., Piao, S., and McEnery, A. M. 2004. The UCREL Semantic Analysis System. Retrieved on 21st January, 2018. http://eprints.lancs.ac.uk/1783/

Richard Socher, John Bauer, Christopher D. Manning and Andrew Y. Ng 2013. Parsing with Compositional Vector Grammars. In ACL. Vol 1, pp. 544-465.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell III, J. T., and Johnson, M. 2002. Parsing the Wall Street Journal Using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In Proceedings of the 40th ACL, pp. 271-278.

Ristad, E.S., 1989. Computational Structure of GPSG Models: Revised Generalized Phrase Structure Grammar. MIT Artificial Inellengence Laboratory. A.I.T.R.No. 1170.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. In Proceedings of ACL.

Sag, I. 1995. HPSG: Background and Basics, Stanford, CSLI; Retrieved on 1st November 2017. http: //hpsg.stanford.edu/hpsg.

Sankaravelayuthan, R., Anandkumar, M., Dhanalakshmi, V. and Mohan Raj, S.N., 2019. A Parser for Question-answer System for Tamil. QA System Using DL, 229, p.230.

Sankaravelayuthan, Rajendran. 2017. MULTI WORD EXPRESSIONS IN TAMIL. https://www.researchgate.net/publication/316039435_MULTI_WORD_EXPRESSIONS_IN_TAMIL

Santorini, B. 1990. Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd revision). In Technical Reports (CIS), pp. 570.

Sarveswaran, K. 2022. A Deep syntactic parser for the Tamil language [Doctoral dissertation, University of Moratuwa]. Institutional Repository University of Moratuwa. http://dl.lib.uom.lk/handle/123/21176

Sarveswaran, K. and Dias, G., 2020. ThamizhiUDp: A dependency parser for Tamil. arXiv preprint arXiv:2012.13436.

Sekine, S., 1997, March. The domain dependence of parsing. In Fifth Conference on Applied Natural Language Processing. pp. 96-102.

Sells, P. 2013. Lexical-Functional Grammar. In M. den Dikken (Ed.), The Cambridge Handbook of Generative Syntax. pp. 162–201. chapter, Cambridge: Cambridge University Press.

Seraji, M., Jahani, C., Megyesi, B., & Nivre, J. 2014. A Persian Treebank with Stanford Typed Dependencies. In the 9th International Conference on Language Resources and Evaluation (LREC), pp. 796-801.

Stanford Parser. Retrived on 1st June, 2018. https://nlp.stanford.edu/software/srparser.html.

Steedman, M. 1996. A Very Short Introduction to CCG. Unpublished paper. Retrieved on 21st September, 2018. http://www.coqsci.ed.ac.uk/steedman/paper.html

Steedman, M., and Baldridge, J. 2011. Combinatory Categorial Grammar. In Non‐Transformational Syntax: Formal and Explicit Models of Grammar. pp. 181-224.

Stolcke, A. 1995. An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities. In Computational linguistics, Vol.21. No.2, pp. 165-201.

Sureka, K., Srinivasagan, K. G., and Suganthi, S. 2014. An Efficiency Dependency Parser Using Hybrid Approach for Tamil Language. Retrieved on 20th December, 2017. https://arxiv.org/abs/1403.6381

Tammewar, A. 2015. Cost Effective Dependency Parsing for Indian Languages.Hyderabad: IIIT. MS dissertation.

Tandon, J., and Sharma, D. M. 2017. Unity in Diversity: A Unified Parsing Strategy for Major Indian Languages. In Proceedings of the Fourth International Conference on Dependency Linguistics, pp. 255-265.

Tapaswi, N., Jain, S. and Chourey, V. 2012, May. Parsing sanskrit sentences using lexical functional grammar. In 2012 International Conference on Systems and Informatics (ICSAI2012) (pp. 2636-2640). IEEE.

Taylor, A., Marcus, M., and Santorini, B. 2003. The Penn Treebank: An Overview. In Treebanks. Dordrecht: Springer, pp. 5-22.

Titov, I., and Henderson, J. 2010. A Latent Variable Model for Generative Dependency Parsing. In Trends in Parsing Technology. Netherland: Springer, pp. 35-55.

Tse, D. and Curran, J.R., 2010, August. Chinese ccgbank: extracting ccg derivations from the penn chinese treebank. In Proceedings of the 23rd international conference on computational linguistics. Coling 2010. pp. 1083-1091.

UCREL Parsing Tagset. Retrieved on 1st June, 2018. http://ucrel.lancs.ac.uk.

Universal Dependency (UD) Tagset. Retrieved on 1st June, 2018. http://universaldependencies.org

Vijay Sundar Ram and Sobha Lalitha Devi. 2021. Dependency Parsing in a Morphological rich language, Tamil. In Proceedings of the First Workshop on Parsing and its Applications for Indian Languages, pages 20–26, NIT Silchar, India. NLP Association of India (NLPAI).

Parsing in Indian Languages: With Special Reference to Tamil

The State-of-the-Art

Keerthana B. and Parameswari K.
tulips91@gmail.com, parameshkrishnaa@gmail.com

ABSTRACT: Parsing is the task of assigning syntactic/syntactico-semantic roles for sentences by segmenting sentences into relevant processing units. The tool used for the automation of such process is a parser, a major module in building Natural Language Processing (NLP) applications like Machine Translation systems. The paper aims to provide the current scenario of parsing in Indian languages in general and Tamil, in particular, focusing on the grammar formalisms and annotation schema available. It also attempts to study existing research works to gain an overall understanding of the current parsing techniques and the state-of-the-art of research in Indian NLP in the global scenario.

1. Introduction

The parser is an automated NLP tool used for syntactic/syntactico-semantic analysis of sentences. It processes the input sentences based on the grammar formalism followed in implementation and produces output as constructed parse trees. Building a parser is a challenging task as it involves in handling structural ambiguities in languages. Structural ambiguities are realized due to two different factors; (i) attachment ambiguity and (ii) coordination ambiguity. They are illustrated in examples (1) and (2) respectively:

(1) I saw a girl with the telescope.
(2) Old men and women

In the example (1), the attachment of noun phrase (NP) (*the telescope)* shows an ambiguity as it can be potentially attached with the subject NP (*I)* or with the object NP (*the girl)*. In the example (2), the coordination ambiguity is shown where the adjective *old* may have interpreted to be coordinated with either *men* or *women.*

The parser attempts to resolve such structural ambiguities based on various factors such as morphological, syntactic, semantic, contextual and discourse knowledge of a language. Once ambiguities are identified, the parser attempts to choose rightly parsed output with the given knowledge. Hence, building a parser with an effective algorithm is desirable for an efficient disambiguation process.

The focus of this paper is to discuss the state-of-the-art of parsers in Indian languages with a special reference to Tamil, a Dravidian Language. It includes discussions on grammar formalisms, annotation schema and techniques followed in implementing parsers for the

above-said languages focusing research taken place since 2009 till today. The paper also discusses the suitable tagset for Tamil by considering specific language features and grammar formalisms.

The paper is divided into six sections discussing the following topics:

(i) Grammar formalisms in Parsing
(ii) Parsing technique
(iii) Types of parser
(iv) Annotation schema
(v) Parsers: a review
(vi) A discussion on a suitable model for Tamil

2. Grammar Formalisms in Parsing

Linguistic understanding and selecting suitable grammar formalism are considered to be the base for building an effective parser for any language. A number of grammar formalisms are available for analysing languages and some of them are also adapted in building computational parsers. This paper discussed formalisms such as (i) Generalized Phrase Structure Grammar (ii) Head-driven Phrase Structure Grammar (iii) Combinatory Categorial Grammar (iv) Lexical-Functional Grammar and (v) Dependency Grammar. This section also discusses general strategies of parsing such as top-down and bottom-up parsing and types of parsing in implementation.

2.1. Generalised Phrase Structure Grammar (GPSG)

GPSG, a constraint-based grammar was developed by Gerald Gazdar in the 1970s with Ewan Klein, Ivan Sag, and Geoffrey Pullum for English (Cf. Gazdar, G., et.al, 1985), deriving from constituency grammar. One of its main goals is to show that natural languages can be expressed in context-free grammars. The analysis of GPSG for example (3) is given below.

(3) He gave the woman the gift

GPSG output:

((NP-he (N)))((VP-gave (V)))((NP- the (DET) woman (N)))((NP-the (DET) gift (N)))

Later, Bahrani, M. et.al. (2011) claimed that GPSG, a unification-based formal theory applies to languages like Persian, French, Chinese and Arabic. It was provided with an instance of Persian output, which had 84.5% parsed output out of 89% accepted sentences among 1200 sentences (with varying contexts), using a hybrid approach and following X- bar theory. Philips J.D. (1992) reasoned out that the annotated representations were not enough for the parser to interpret several hundred rules (especially for constituent order languages) that were formulated by GPSG to

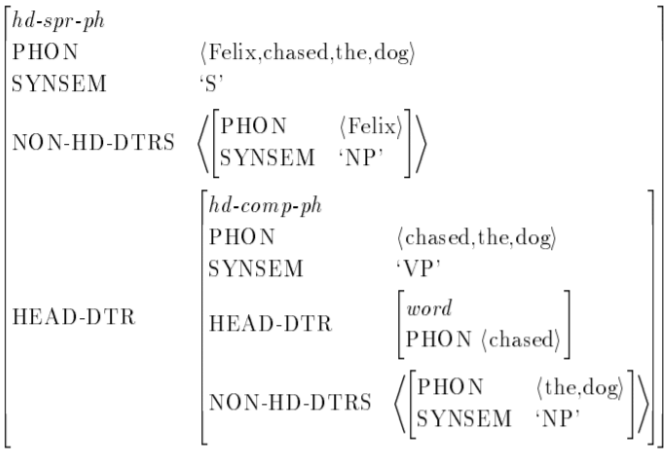
analyse the natural language and as a result, a parsed output of above 90% was unachievable, even after revising the rules of GPSG.

2.2. Head-driven Phrase Structure Grammar (HPSG)

HPSG, the successor of GPSG is a lexical- based, constrained PSG developed by Carl Pollard and Ivan Sag (Cf. Pollard, C., and Sag, I. A. 1994). It deals with the *sign*, taking *words* and *features* as sub-types of the sign. Words are represented by PHON and SYSTEM. These signs and the formulated rules are together called feature structures. The structure of HPSG output for the example (4) is seen in figure 1 below:

(4) Felix chased the dog

HPSG output (extracted from Sag, I. A., 1995:15):



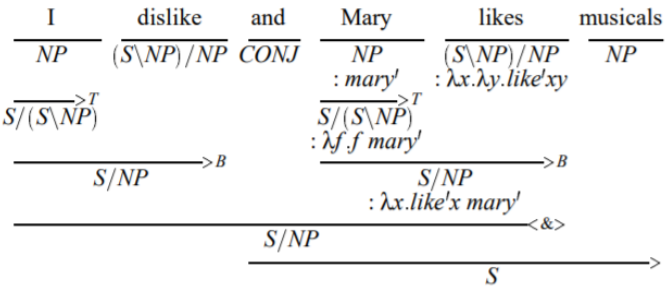
Levine, R. D. and Meurers, W. D. (2006) claimed that HPSG is suitable to apply for languages such as Romance languages, Slavic languages, German, Japanese, Welsh, English, Korean and Warlpiri .The theory was applied in '*Enju*', an English probabilistic HPSG parser developed by Tsujii Laboratory (University of Tokyo), Japan, which was derived from Penn Treebank. Proudian, D. and Pollard, C. (1985) have stated that HPSG would be comparatively advantageous as it could give a good accuracy because importance is given to the heads of the phrases unlike GPSG.

2.3. Combinatory Categorial Grammar (CCG)

The onset of CCG in computational linguistics was in the late 1970s and early 1980s, where the categorial grammar was extended with functional operators like the functional composition, substitution, etc. This grammar focused on grammatical constituents which were differentiated by syntactic types, identifying them either as a function from arguments or as an argument (Steedman, M., and Baldridge, J. 2011). The example (5) is taken from Mark Steedman (1996:4):

(5) I dislike and Mary likes musicals

CCG Parser Output:



CCG is also used in Penn Treebank even though it has a huge number of categories. The trained corpus has 1207 categorial lexicons which are compared with 48 POS tags of Penn Treebank. On the other hand, CCG has a very few grammatical rules to accommodate such a huge number of categories, resulting in a less over-generating grammar. The final recall of the system is 89.9% against the trained data (Hockenmaier, J., and Steedman, M., 2002:342).

2.4. Lexical Functional Grammar (LFG)

LFG is first published by Joan Bresnan (1982), which addresses the mechanisms to extract grammatical relations from a sentence in a positional language such as English. Neidle (1994) claims that the main focus of LFG is on syntax even though the grammar extends its relation with morphology and semantics. LFG is being represented in constituent and functional structure as seen below:

LFG output for the example (3):

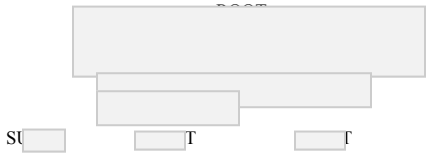
SUBJ	PRED	'he'
	NUM	sg
	PERS	1
	DEF	+
PRED	'give <agent,goal,patient>'	
	SUBJ,OBJ2,OBJ1	
TENSE	past	
OBJ2	PRED	'the woman'
	NUM	sg
	PERS	3
	DEF	+
OBJ1	PRED	'the gift'
	NUM	sg
	PERS	3
	DEF	+

LFG is used in parsing Wall Street Journal (WSJ) by Stefan Riezler, et.al. (2002). The evaluation output is shown as 74.7% accuracy for full parses and 25.3% for fragment parses. In this process, Brown corpus is gold standardly annotated and trained. The major advantage found in LFG is that the mismatch between the surface structure and the deep argument structure as discussed in Chomskyan framework is not found here. However, LFG does not deal with lexical ambiguity, optional theta roles, adjuncts, and mapping from grammatical relations to theta- roles (Bharati, A., et.al, 1995).

2.5. Dependency Grammar (DG)

Dependency approach (both projective and non-projective) is one of the approaches to automatically parse the natural language, following the grammatical tradition of dependency grammar, tracing back to Panini's grammar. While old school of thought is still in practice, the modern thought was proposed by a French linguist Lucien Tesnière during post-1950s, which attempted to capture the grammar of all typologies including Indian languages' typology. The dependency grammar structure represents the relation between the head and its dependents through directed arcs and the functional categories in the form of arc labels. Content words are marked by dependency relations; functional words attach to the content words they modify and punctuation attach to the head of the phrase/ clause. The parse is a tree, where the nodes stand for the words in an utterance and the link between the words represent the relation between the pair of words. Such dependencies can either be argument dependencies (subject, object, indirect object, etc.) or modifier dependencies (determiner, noun modifier, verb modifier, etc.). The dependency parser output for the example (3) is given below.

Dependency parser output:



He gave the woman the book .

In the above output, the verb root is considered as the head and other noun phrases are related with the verb as its dependents with various *kāraka* roles (i.e SUBject, OBJect, IndirectOBJect). The dependency grammar is widely used in building parsers for Indian languages (*see* section 6.2 for more information).

3. Parsing Technique

The parsing system is implemented in two different styles: (i) top-down parsing (ii) bottom-up parsing.

3.1. **Top-down parsing**:

Top-down parsing attempts to construct a parsed tree for the input from the root (top) to the leaves (bottom), where the transitions of tokens are seen from left to right, attempting to resolve ambiguities by changing the rules of right hand side. The major advantage of such systems is that it never wastes time in validating trees that would not lead to S (root) but the negative aspect is that it processes output before examining the input (Cf. Jurafsky, 2000: 356-359). This type of parsing uses Context-free grammar, which is a set of rewriting rules. Recursive descent parsers and LL[1] parsers are examples of such kind of parsing.

3.2. **Bottom-up parsing:**

Bottom-up parsing constructs a parsed tree from the leaves to the root, i.e. from bottom to top. The positive aspect of such systems is that it never suggests a tree that is not grounded to input but never reaches to the root, S. Grammar formalisms such as GPSG, HPSG, CCG, LFG, and DG are applied through bottom-up parsing. LR[2] or shift reducing parsers like MALT[3] are examples for such parsing.

4. Types of Parser

The parser is implemented in various ways with the suitable grammar formalism. In this section the four major types of parsers are discussed.

4.1. Rule-based Parser

Rule-based parsing uses pre-written rules to describe the data. Using rule-based parsing, the main functor-argument relations are obtained. Grammar-driven dependency parsing is a type of rule-based parsing, which is formed from the combinations of context-free and constraint -based Dependency Parsing (DP), which is well defined by the grammar of the language. An input sentence of the language is validated, only when it is accepted by the grammar of the language as the formal language is vital in this approach (Cf. Kübler, S., Ryan McDonald, Joakim Nivre and Graeme Hirst, 2009: 64-70). Weighted Constraint Dependency Grammar (WDCG) is another example of rule-based parsing (Krivanek, J., and Meurers, D., 2013).

4.2. Statistical Parser

In Statistical parser, grammar rules are associated with probability of a complete parse of a sentence. In building statistical parsing, the commonly used grammar formalism is probabilistic context-free grammars (PCFG). Data-driven dependency parsers (Nivre, 2006) follow machine learning approach and thus, it is widely used in statistical models. Any sentence or phrase given

[1] LL- Left to right, Left most derivation type
[2] LR- Left to right, Right most derivation type
[3] MALT- MaltParser, developed by Johan Hall Jens Nilsson and Joakim Nivre at Växjö University and Uppsala University, Sweden is a system for data-driven dependency parsing, which can be used to induce a parsing model from treebank data and to parse new data using an induced model. (http://www.maltparser.org).

as input is considered as a valid grammatical sentence and parsed. Data-driven dependency parsing is sub-categorized into two types, transition-based dependency parsing and graph-based dependency parsing. MALT is the best example for statistical parser. However, statistical parser has drawbacks like lack of lexical conditioning and poor independence assumptions but can be improved by annotating bigger data (Jurafsky, 2000).

4.3. Hybrid Parser

A combination of rule-based and statistical parsing results in hybrid parsing, where the rules are applied to sentences after the machine learning. It gives better accuracy than the rule-based and probabilistic/stochastic models as both these models are inbuilt in this system. The requirement includes fully annotated treebank for probabilistic parsing and fully developed rules for the second phase of implementation (Cf. Kilian A. Foth, Wolfgang Menzel, 2006).

4.4. Neural Network Based Parser

Neural network based parser works in dependency approach in both transition-based and graph-based dependency parsing. Yet, commonly found neural network parsers use transition based dependency parsing, where the parser is powered by neural network[4]. The input word embeddings are represented in vectors as shown in Figure (1).

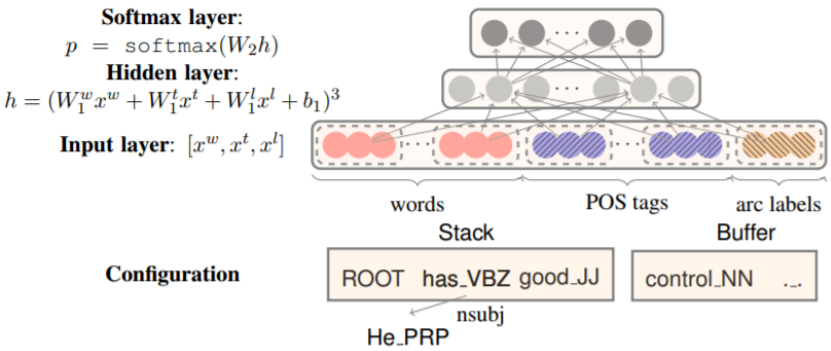


Figure 1: Neural Network Schema (Danqi Chen and Christopher D. Manning, 2014).

5. Annotation Schema

Annotation schema is used in parsing to have a uniform pattern in marking features of a big data. There are many annotation schema with tagset and guidelines are available in building parsers. In this section, the tagsets such as Penn tagset (Santorini, B., 1990), UCREL parsing tagset

[4] Neural Network is an information processing paradigm, which is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems (https://www.doc.ic.ac.uk).

(ucrel.lancs.ac.uk), Prague tagset (ufal.mff.cuni.cz), Stanford tagset (De Marneffe, M. C., and Manning, C. D., 2008), Chinese Dependency tagset (Liu, H., and Huang, W., 2006), Anncorra tagset (Bharati, A., et al. 2009), Universal Dependency (UD) tagset (universaldependencies.org) are discussed. Among these tagsets, Anncorra tagset (Pāninian framework) and Universal Dependency tagset are taken into a detailed discussion, as it is mostly used in implementing parsers for Indian languages.

5.1. Penn Tagset

Penn Treebank (1989-1996), developed by University of Pennsylvania contains the POS tagged and syntactically bracketed forms of Brown corpus and Wall Street Journal. The Treebank has annotated 7 million words of part-of-speech tagged text, 3 million words of skeletally parsed text, over 2 million words of text parsed for predicate argument structure, and 1.6 million words of transcribed spoken text annotated for speech disfluencies (Cf. Taylor, A., M. Marcus, and B. Santorini, 2003). It has 42 fine grained POS tags, 8 chunk tags, and 9 coarse grained relation tags, which are used in parsing. The major aim in introducing the tagset was to reduce the lexical and syntactic redundancy (Cf. Santorini, B., 1990).

5.2. UCREL Parsing Tagset

The UCREL tagset, developed by Lancaster University is used in semantic analysis systems of English. It has 21 coarse-grained discourse tags and 232 fine-grained semantic tags (Paul Rayson, et.al., 2004). The accuracy of the manually tagged system developed by Paul Rayson, Dawn Archer, Scott Piao and Tony McEnery had a precision of 91%.

5.3. Prague Dependency Tagset

Prague Dependency Treebank was developed by Prague School of Functional and Structural Linguistics. The project began in 1995 with the notion of following Praguian dependency tradition and building a Treebank similar to Penn Treebank. They have collected the database from the Czech National Corpus (Charles University, under the guidance of F. Čermák co-joint with other research centres/institutions), and developed a three-layer system of tags: morphemic, syntactic at analytical level, and syntactic at tectogrammatical level (The Prague Dependency Treebank 3.0.). They had developed 68 fine-grained POS tagsets. (https://ufal.mff.cui.cz/).

5.4. Stanford Dependency Tagset

Stanford Dependency tagset was developed by a group of people from Linguistics and Computer Science as a part of AI lab in 2005 for English. It was later extended to Chinese, Italian, Bulgarian and Portuguese. The main was to have a simple representation of the analysed sentences which could be used by commons to extract word relations. The present Stanford Dependency Treebank has an approximate count of 50 relation tags (Cf. De Marneffe, M. C., and Manning, C. D., 2008). Most of these tags are also seen in Universal Dependency tagset.

5.5. Chinese Dependency Tagset

Chinese Dependency Treebank 1.0 was released on May 2012 in Harbin Institute of Technologys Research Center for Social Computing and Information Retrieval (HIT-SCIR) for Mandarin Chinese, Chinese. It was developed by Wanxiang Che, Zhenghua Li, Ting Liu. It contains 49,996 Chinese sentences with 902,191 words, which were sourced from Peoples Daily newswire stories (1992-1996) and annotated with syntactic dependency structures. The data is provided in the format of CoNLL-X and in UTF-8 and has 13 word class tags and 34 fine grained dependency tags (Cf. Liu, H., and Huang, W., 2006).

5.6. Anncorra Tagset

Annotated Corpora (Anncorra) is developed based on the Pāninian Dependency grammar with *kāraka* and non-*kāraka* relations aiming at a uniform representation of annotated corpus of Indian languages (Bharati, A., et.al, 2002). It is developed for parsing Hindi sentences and the tags are given accordingly. Later, the same guidelines were adapted for other Indian languages (Marathi, Urdu, Bengali, Kannada, Telugu, Tamil, and Malayalam) (Cf. Tandon, J. and Sharma, D. M., 2017). It was even used by Amita, A. J. (2015) for English, in which HyDT annotation scheme and hybrid approach (statistical+ rule based) were used for parsing 2000 words.

The 19 fine-grained *kāraka* relations that are included in the Anncorra tagset are k1 (*karta* ‘doer/agent/ subject’), pk1 (*prayojaka karta* ‘causer’), jk1 (*prayojya karta* ‘causee’), mk1 (*madhyastha karta*‘mediator-causer’), k1s (*karta samanadhikarana*- ‘noun complement of *karta*’), k2 (*karma* ‘object/patient’), k2p (Goal, Destination), k2g (secondary *karma*), k2s (*karma samanadhikarana* ‘object complement’), k3 (*karana* ‘instrument’), k4 (*sampradana* ‘recipient’), k4a (*anubhava karta* ‘Experiencer’), k5 (*apadana* ‘source’), k5prk (prakruti apadana ‘source material’), k7t (*kAlAdhikarana* ‘location in time’), k7p (*deshadhikarana* ‘location in space’), k7 (*vishayadhikarana* ‘location elsewhere’), k7a (according to) and k*u (*sAdrishya* ‘similarity/comparison’).

The 25 fine-grained non-*kāraka* relations include genitive case, adverbial and adjectival relations. It includes, r6 (*shashthi* ‘genitive/possessive’), r6-k1, r6-k2 (*karta* or *karma* of a conjunct verb (complex predicate)), r6v (*kA* ‘relation between a noun and a verb), adv (*kriyAvisheSaNa* 'manner adverbs'), sent-adv (Sentential Adverbs), rd (direction), rh (*hetu* ‘reason’), rt (*tadarthya* ‘purpose’), ras-k* (*upapada sahakArakatwa* ‘associative’), ras-neg (Negation in Associative), rs (noun elaboration), rsp (address terms), nmod__relc, jjmod__relc, rbmod__relc (relative clauses, *jo-vo* constructions), nmod (participles etc. modifying nouns), vmod (verb modifier), jjmod (D-Rel modifiers of the adjectives), pof (part of units such as conjunct verbs), ccof (co-ordination and sub-ordination), fragof (Fragment of), enm (enumerator), rsym (ag for a symbol) and psp_cl (relation between clause and postposition following that clause).

5.7. Universal Dependency (UD) tagset

Universal Dependency is a cross-linguistic project, built with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective (http://universaldependencies.org). The idea of annotation scheme has been taken from Stanford dependencies, Google universal part-of-speech tags and the Interset interlingua for morphosyntactic tagsets in 2013 (McDonald et al., 2013). The parser has a lesser number of modules (Preprocessing (transliteration, sentence segmentation, tokenization); M-layer annotation (positional tagging) and A- layer annotation (dependency annotation), making it universal for any language typology. Indian languages like Hindi, Marathi, Sanskrit, Tamil, Telugu, and Urdu are included in the existing UD Treebank and Kannada and Pnar are upcoming languages listed in the UD website.

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

Figure 2: Universal Dependency tagset (http://universaldependencies.org)

The figure 2 lists 37 coarse grained UD tags including *acl* (clausal modifier of noun (adjectival clause)), *advcl* (adverbial clause modifier), *advmod* (adverbial modifier), *amod* (adjectival modifier), *appos* (appositional modifier), *aux* (auxiliary), *case* (case marking), *cc* (coordinating conjunction), *ccomp* (clausal complement), *clf* (classifier), *compound* (compound), *conj* (conjunct), *cop* (copula), *csubj* (clausal subject), *dep* (unspecified dependency), *det* (determiner), *discourse* (discourse element), *dislocated* (dislocated elements), *expl* (expletive), *fixed* (fixed

multiword expression), *flat* (flat multiword expression), *goeswith* (goes with), *iobj* (indirect object), *list* (list), *mark* (marker), *nmod* (nominal modifier), *nsubj* (nominal subject), *nummod* (numeric modifier), *obj* (object), *obl* (oblique nominal), *orphan* (orphan), *parataxis* (parataxis), *punct* (punctuation), *reparandum* (overridden disfluency), *root* (root), *vocative* (vocative) and *xcomp* (open clausal complement).

Apart from these listed universal tags, there are 198 language specific tags that are used in various languages parsing system. For instance, in Tamil, experiencer subjects in dative subject construction require a different tag as it is inflected with the non-nominative case and verbs do not agree with the so-called subject. The experiencer subjects are given with the tag 'nsubj:nc' i.e.. non-canonical subjects/dative subjects.

(6) *eṉa-kku* *paṭam* *pār.kk-a* *vēṇṭum*
I-DAT movie see-INF want
'I want to see a movie'

In the example (1), the dative-marked subject *eṉa-kku* 'I-DAT' is given with the tag 'nsubj:nc'.

The table (1) provides the statistics of Indian language in UD project.

S.No.	Language	No. of annotated sentences	No. of UD tags	No. of language-specific tags
1	Hindi	17,647	13	3
2	Urdu	5130	16	1
3	Telugu	1328	14	11
4	Tamil	690	14	3
5	Marathi	466	15	8
6	Sanskrit	225	14	11

Table (1): Statistics of Indian language in UD project.

6. Parsers: A Review

Parsers are being implemented worldwide for various languages. This section deals with a review of parsing in the following languages:

(i) World languages
(ii) Indian languages
(iii) Tamil

6.1. World Languages

A parsing algorithm recorded to be the earliest was proposed by Yngve (1955). Yet, most of the parsers were developed in early 1990s. Such implemented parsers for world languages include:

- Collins's (1999) statistical parser for Czech using Prague Dependency Treebank
- Eugene Charniak's (2000) maximum-entropy parser for English
- Bikel and Chiang's (2000 first statistical model on Chinese Treebank
- DeSR, developed by Yamada and Matsumoto (2003) for English
- Dubey and Keller's (2003) proposal of a probabilistic parsing for German
- Stanford parser (2003), a statistical parser, using lexicalized PCFG (Probabilistic Context-Free Grammar), developed by Dan Kleinbeing for English and further extended to Arabic, Chinese, French, German and Spanish
- A probabilistic parser with supervised learning based on PCFG for English (Collins, 1997)
- Robust Accurate Statistical Parsing (RASP) System, a hybrid domain independent English parser (Cf. Briscoe, et.al, 2006)
- MALT (2007) and MST (2005) (developed by Johan Hall, Jens Nilsson and Joakim Nivre at Växjö University and Uppsala University, Sweden), a transition based parser
- ISBN (Incremental Sigmoid Belief Networks), a trainable dependency parser (Cf. Titov, I. and Henderson, J., 2010)
- Carnegie-Mellon's Link Grammar parser, built for English, Arabic, Russian and Persian
- Seraji, M., Jahani, C., Megyesi, B., and Nivre's (2014) work on Persian by obtaining the data from large-FARSDAT
- Universal Dependencies (UD) (McDonald et al., 2013), a project developed by Joakim Nivre, which is involved in developing a cross-linguistic study, maintaining a treebank annotation for 60 languages with 102 treebanks

6.2. **Indian Languages**

Parsing is one such area, which has to be explored in depth for Indian languages. Some of the Indian languages including, Hindi, Urdu, Telugu, Kannada, Tamil, Bengali, Marathi, and Assamese have delved into area of parsing, which are still work in progress. (Cf. Monika T. Makwana and Deepak C. Vegda, 2015). In fact, Tandon, J., and Sharma, D. M. (2017) has come up with a unified strategy for parsing Indian languages using Pāninian framework. Researches related to cost-effective methods of building dependency parser for Indian languages are also in the current trend (Cf. Tammewar, A. 2015). The list of implemented parsers in Indian languages between 2009 and 2018 is discussed below:

- Nivre (2009) optimizied MALT parser for Hindi, Bengali and Telugu. With coarse-grained tagset, the respective accuracies are 81.1%, 79.6% and 63%. But, when fine-grained tagset is used, it had lower accuracies, i.e. 75.3%, 72.9% and 58.5%.
- Hindi, Bengali and Telugu sentences are tested with MALT and MST (data-driven parsers) by Bharat Ram Ambati, et.al. (2009), where MALT has a better performance than MST. The report has a final average score as 88.43%, 71.71% and 73.81% respectively.
- A bidirectional dependency parser for Hindi, Bengali and Telugu is proposed by Prashanth Mannem (2009), which shows the accuracy of 71.63%, 59.86% and 67.74% respectively

when run with test data. The same data has better accuracies with the coarse-grained tagset, 76.90%, 70.34% and 65.01% respectively.

- A constraint based dependency parsing system for Bengali with Pāninian Grammar formalism is proposed by Sankar De, et.al. (2009), which is trained with 1000 annotated sentences, and evaluated with 150 sentences. It has the accuracy of 79.81%, 90.32% and 81.27% for labelled attachments (LAS), unlabelled attachments (UAS) and label scores (LS) respectively.
- Aniruddha Ghosh, et.al. (2009) trains Bengali data using CRF and was implemented using rule-based algorithm. It results in 74.09% (LAS), 53.09% (UAS) and 61.71% (LS).
- Sanjay, et.al. (2009) has run Bengali sentences on a data-driven parser and hybrid parser. The wrongly annotated sentences are given rules to improve the accuracy. A special look at subject, object, location and relation is observed.
- Rahman, Mirzanur, et.al. (2009) analyse the issues in areas of parsing Assamese sentences when tagged with 7 tags based on CFG formalism. Later, rules are developed accordingly and algorithms are modified from Earley's Algorithm to solve those issues.
- A constraint-based Hindi dependency parsing system with the accuracy of 62.20% (LAS) and 85.55% (UAS) is implemented by Meher Vijay Yeleti and Kalyan Deepak (2009).
- Bharat Ram Ambati, et.al. (2010) analyse the role of linguistic features in data-driven dependency parsing for Hindi and found that accuracy gain is seen when adding morphosyntactic features like case and TAM features. They had finally gained 2% accuracy (76.5% in total) after combining morph features from two different parsers.
- Antony P.J. (2010) has developed a statistical syntactic Kannada parser using Penn Treebank with 1000 POS tagged sentences using SVM POS tagger. It is implemented using supervised machine learning and is evaluated using SVM algorithms. As a result, they claim to have good accuracy.
- B.M.Sagar (2010) has developed a CFG for Kannada parser and finally proposes that top-down parser is best suited for Kannada.
- Navanath Saharia, et.al. (2011) have used CFG to parse the simple sentences of Assamese, which is not implemented.
- B.Venkata S. Kumari, et.al. (2012) use a combination of MALT and MST parsers which shows LAS 90.66% for gold standard and 80.77% for automatic tracks.
- Karan Singh, et.al. (2012) propose a two-stage approach for Hindi Dependency Parsing using MALT parser. Their system has a record of 90.99% (LAS) for the gold standard.
- Uma Maheshwar Rao G., K. Rajya Rama, A. Srinivas (2012) has worked on Dative case towards building a parser. Various functions of the dative marker is discussed and a flowchart is developed to build a robust parser for Telugu.
- Sambhav Jain, et.al. (2013) has added the ontological features to Hindi dependency parser which added the accuracy improvement of 1.1% (LAS) for 1000 sentences and 0.2% (LAS) for 13371 sentences.

- A Lexicon parser for Devanagiri script (Hindi) is developed by Swati Ramteke, et.al. (2014), which generated semantic parsed trees with an accuracy of 89.33% when run with unambiguous sentences. Rule-based approach was used to resolve the lexical ambiguities.
- Arpita Batra, Soma Paul, and Amba Kulkarni (2014) had worked on the constituency analysis for Hindi using four approaches. Adjacency global, adjacency greedy, dependency global and dependency greedy were applied for 2322 sequences of words. Applying all these approaches, 92.85% (using global dependency algorithm and syntactic rules) accuracy was obtained.
- Dhanashree Kulkarni, et.al. (2014) has taken up CFG as the grammar formalism and used the same in Top-Bottom and Bottom-Top parser for Marathi. The final outcome of the paper was to develop (computerized) grammar checking for Marathi text from CFG perspective.
- A Combinatory Categorical Grammar (CCG) Telugu treebank is created using CCG lexicon and dependency Treebank and it is tagged with CCG supertags as features to Telugu dependency parser. An improvement of 1.8% in UAS and 2.2% in LAS (especially on verbal arguments) was observed when implemented using MST parser (Cf. Kumari, B. and Rao, R. R., 2015).
- Telugu Dependency parser, developed by Nagaraju, G. et.al. (2016) have used bottom-up parser and parsed 200 Telugu sentences using *kāraka* relations. Out of 200 sentences, they have obtained 178 correct parsed sentences. As a whole, 880 words were correctly tagged and 140 were incorrect and thus, they claim the precision to be 99.
- 'Improving Transition-Based Dependency Parsing of Hindi and Urdu by Modeling Syntactically Relevant Phenomena', by Bhat, R. A., Bhat, I. A., and Sharma, D. M. (2017) have used *kāraka* and non-*kāraka* relations and annotated the inter-chunk dependencies manually. They have implemented in transition-based dependency parser with syntactic features and obtained an accuracy of 87.82% in trained set and 87.72 in test data of LAS.

6.3. Tamil

Tamil, belonging to Dravidian language family, is morphologically rich. It has a (S)OV word order with agglutinative morphology. Hence, building a parser for Tamil is a challenging task. This section lists the Tamil parsers with grammar formalisms and techniques used in their respective parsers.

- Hybrid approach combining PSG and DG with Lexicalized and Statistical Parsing (LSP) is used by Selvam, M., Natarajan, A. M. and Thangarajan, R. (2008) with 500 tags and 31 dependency relations on Tamil. 3261 sentences with 51026 words are used and as a result, 73% accuracy in trained data and 65% accuracy in test data with just 600 trained sentences were obtained. The lacuna is seen in their choice of their tagset which had 500 tags.
- Loganathan Ramasamy and Zdeněk Žabokrtský (2011) have done initial experiments with Tamil dependency parsing using rule-based approach (with an accuracy of 79% (LAS) in the trained data and 61% with the test data) and corpus based approach (with an accuracy of

75%. Finally, it is concluded that both the approaches have failed in identifying coordination nodes.

- Universal Dependency has extended its system to Tamil, by developing a Tamil Treebank (from Prague dependency Treebank) (Cf. Ramasamy, Loganathan and Zdeněk Žabokrtský, 2012), which has universal tagsets and just involves three processes: Pre-processing (transliteration, sentence segmentation, and tokenization); M-layer annotation (positional tagging) and A- layer annotation (dependency annotation) with 217 distinct tags (including all 9 positions). 96% of the test data was unambiguous; 3% was ambiguous with 2 tags and tokens with 3-4 tags were just 1% which is negligible. Altogether, 21 dependency relations are used for labeling edges. It has an accuracy of 69% when trained with 690 sentences.
- A Tamil syntactic parser, proposed by K. Sureka, Dr. K. G. Srinivasagan and S. Suganth (2014) works on dependency grammar and follows hybrid approach, with clause boundary identifier. After adding the module, the result obtained is that out of 150 sentences, 120 are parsed correctly.
- Vigneshwaran (2017) has worked on Tamil parsing based on cognitive grammar as the theoretical grammar and Pāninian framework as the computational grammar. The main argument revolves around parsing Tamil sentences at discourse level, as it claims that sentential analysis is not enough to get an idea of the complete context of the text.

7. A Discussion on a Suitable Model for Tamil

Tamil, a Dravidian language is morphologically rich, when compared to Indo-Aryan languages, hence building a syntactic parser becomes a complex task. A range of language-specific annotation schema is required to cover a good range of sentence structures for a better accuracy.

The dependency approach is considered to be the best suitable model for Tamil over other formalisms available. It has a better reach than the Constituency approach as it accommodates maximum typologies of languages, making it universal by accommodating a maximum number of languages. Phrase Structure Grammar and other formalisms are related to constituency parsing have failed to look at the semantics, which has led to multiple drawbacks including no mechanisms for resolving structural ambiguities, not able to map theta roles etc. Moreover, constituency parsing's average number of nodes is twice as the number of dependency nodes. Thus, automated DP is faster than constituency parsing.

When Indian languages with different typologies try to adapt the Pāninian framework which was actually designed for Sanskrit, adding the language specific features may gain good accuracy. For instance, Anncorra tagset, designed based on *kāraka* systems may not accommodate a range of sentence structures that are available in Tamil. Similarly, Universal Dependencies have a coarse grained tagset for all languages, though some language-specific tags are introduced to some languages. Whereas, Tamil needs more tags to develop a well-annotated corpus.

As seen in reviews of parsers in different languages, hybrid parsing is considered to have advantageous than other methods as it combines multiple parsing strategies like rule-based and supervised probabilistic parsing. Here, the rules and machine learning algorithms are blended to ensure the effectiveness of the output. Thus, it would give a better output than the standalone rule-based or probabilistic models for any language.

8. Conclusion

The paper has listed a number of grammar formalisms in parsing, parsing techniques, types of parser, and annotation schema that are available for languages. A number of implemented parsers in Indian languages with a special reference to Tamil are discussed. The conclusion inferred from a discussion on a suitable model for Indian Languages is that both UD and Anncorra guidelines have its respective drawbacks. Developing language specific tags would improve the accuracies in both the cases. Implementing well-annotated corpus in Hybrid system would give better accuracies for Tamil.

References

Ambati, B. R., Gadde, P., & Jindal, K. 2009. Experiments in Indian Language Dependency Parsing. In the *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, pp. 32-37.

Amita, A. J. 2015. An Annotation Scheme for English Language Using Paninian Framework. *IJISET*, Vol. 2, pp. 616-619.

Bahrani, M., Sameti, H., and Manshadi, M. H. 2011. *A Computational Grammar for Persian Based on GPSG*. Retrieved on 21st November, 2017. https://link.springer.com/article/10.1007/s10579-011-9144-1

Bresnan, J. 1982. Control and complementation. In *Linguistic inquiry*. Vol. 13. No. 3, pp. 343-434.

Bharati, A., Sangal, R. 1993. Parsing Free Word Order Languages in the Paninian Framework. *In: Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, pp. 105–111.

Bharati, A., Sangal, R., Chaitanya, V., Kulkarni, A., Sharma, D. M., and Ramakrishnamacharyulu, K. V. 2002. AnnCorra: Building Tree-banks in Indian Languages. In *Proceedings of the 3rd Workshop on Asian language Resources and International Standardization: Association for Computational Linguistics*. Vol 12 (pp. 1-8).

Bharati, A., Sharma, D. M., Husain, S., Bai, L., Begam, R., and Sangal, R. 2009. *Anncorra: Treebanks for Indian Languages, Guidelines for Annotating Hindi Treebank.* Retrieved on 1st December, 2016. http://aclweb.org/anthology/W17-63

Bharati, A., Vineet Chaitanya and Sangal, R. 1995. *Natural Language Processing: A Paninian Perspective.* New Delhi: Prentice-Hall of India.

Bhat, R. A., Bhat, I. A., and Sharma, D. M. 2017. Improving Transition-Based Dependency Parsing of Hindi and Urdu by Modeling Syntactically Relevant Phenomena. In *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, Vol 16, No. 3, pp 17.

Bikel, D. M., and Chiang, D. 2000. Two statistical parsing models applied to the Chinese Treebank. In *ACL,* Vol 12, pp. 1-6.

Briscoe, E, Carroll, J and Watson, R. 2006. *The Robust Accurate Statistical Parsing (RASP).* Retrieved on 20th February, 2018. http://sro.sussex.ac.uk/25679/

Charniak, E. 2000. A maximum-entropy-inspired parser. In *ACL,* pp. 132-139.

Collins, M. (1997, July). Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 8th Conference on European Chapter of the ACL*, pp. 16-23.

Collins, M., Ramshaw, L., Hajič, J., and Tillmann, C. 1999. A statistical parser for Czech. In *ACL,* pp. 505-512.

Dan Klein and Christopher D. Manning. 2003a. Accurate Unlexicalized Parsing. In *Proceedings of the 41st ACL*, pp. 423-430.

Dan Klein and Christopher D. Manning. 2003b. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, Cambridge: MIT Press, pp. 3-10.

Danqi Chen and Christopher D. Manning 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *EMNLP,* pp. 740-750.

De Marneffe, M. C., and Manning, C. D. 2008. *Stanford Typed Dependencies Manual.* California: Stanford University. Technical report, pp. 338-345.

Dhivya, R. 2011. *Dependency Parser for Tamil Using Machine Learning Approach.* Retrieved on 15th November, 2016. http://nlp.amrita.edu:8080/project/mhrd/ms/Tamil/DependencyParser_Dhivya_CEN09009.pdf

Dubey, A., and Keller, F. 2003. Probabilistic Parsing for German Using Sister-Head Dependencies. In *Proceedings of the 41st ACL.* Vol 1, pp. 96-103.

Foth, K. A., and Menzel, W. 2006. Hybrid parsing: Using Probabilistic Models as Predictors for a Symbolic Parser. In *ACL*, pp. 321-328.

Garapati, U. R., Koppaka, R., and Addanki, S. 2012. Dative Case in Telugu: A Parsing Perspective. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*, pp. 123-132.

Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. A. 1985. *Generalized Phrase Structure Grammar*. Cambridge: Harvard University Press.

Hockenmaier, J., and Steedman, M. 2002. Generative Models For Statistical Parsing with Combinatory Categorial Grammar. In *Proceedings of the 40th ACL*, pp. 335-342.

Husain, S. (2009). Dependency Parsers for Indian Languages. In *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing*.

Jurafsky, D. 2000. *Speech and Language Processing*. London: Pearson.

Klein, D., and Manning, C. D. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st ACL.* Vol 1, pp. 423-430.

Krivanek, J., and Meurers, D. 2013. Comparing Rule-Based And Data-Driven Dependency Parsing of Learner Language. In *Computational Dependency Theory*, pp. *258*-207.

Krivanek, J., and Meurers, D. 2011. *Comparing Rule-Based and Data-Driven Dependency Parsing of Learner Language.* Retrieved on 1st May, 2018. http://www.depling.org/proceedingsDepling2011/papers/krivanekMeurers.pdf

Kübler, S., Ryan McDonald, Joakim Nivre, Graeme Hirst. 2009. Dependency Parsing. In *Synthesis Lectures on Human Language Technologies*. California: Morgan and Claypool Publishers. Vol 1. No.1.

Kumari, B., and Rao, R. R. 2015. Improving Telugu Dependency Parsing Using Combinatory Categorial Grammar Supertags. In *ACM Transactions on Asian and Low-Resource Language Information Processing*, Vol 14. No.1, pp. 3.

Levine, R. D., and Meurers, W. D. 2006. Head-Driven Phrase Structure Grammar. In *Encyclopedia of Language and Linguistics*. Vol. 5, pp. 237-52.

Liu, H., and Huang, W. 2006. A Chinese Dependency Syntax for Treebanking. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, pp. 126-133.

Makwana, M.T. and D.C.Vegda. 2015. *Survey: Natural Language Parsing for Indian Languages*. Retrieved on 12th January, 2018. https://arxiv.org/ftp/arxiv/papers/1501/1501.07005.pdf

Mannem, P. 2009. Bidirectional Dependency Parser for Hindi, Telugu and Bangla. *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing.*

Mates, Benson. 1961. *Stoic Logic*. Berkeley: University of California Press.

McDonald, R., Crammer, K., and Pereira, F. 2005. Online Large-Margin Training Of Dependency Parsers. In *Proceedings of the 43rd ACL,* pp. 91-98.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., ... and Bedini, C. 2013. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st ACL.* Vol. 2, pp. 92-97.

Nagaraju, G., Mangathayaru, N., and Rani, B. P. 2016. Dependency Parser for Telugu Language. In *ACM, Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies,* pp. 138.

Neidle, C. 1994. Lexical-Functional Grammar (LFG): In *RE Asher, editor, Encyclopedia of Language and Linguistics*. Oxford: Pergamon Press, pp. *9*-25.

Nivre, J. 2006. Inductive Dependency Parsing. In *Text, Speech and Language Technology.* Netherlands: Springer. Vol 34.

Nivre, J. 2009. Parsing Indian Languages with Maltparser. In the *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, pp. 12-18.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. and Marsi, E. 2007. MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. *Natural Language Engineering*, Vol. 13. No. 2, pp. 95-135.

Phillips, J. D. 1992. A Computational Representation for Generalised Phrase-Structure Grammars. *Linguistics and Philosophy*. Vol.15. No.3, pp. 255-287.

Pollard, C., and Sag, I. A. 1994. *Head-Driven Phrase Structure Grammar*. US: University of Chicago Press.

Prague Tagset. Retrieved on 1st June, 2018. http://ufal.mff.cuni.cz

Proudian, D., and Pollard, C. 1985. Parsing Head-Driven Phrase Structure Grammar. In *Proceedings of the 23rd ACL.* pp. 167-171.

Ramasamy, L., and Žabokrtský, Z. 2011. Tamil Dependency Parsing: Results Using Rule Based And Corpus Based Approaches. In *International Conference on Intelligent Text Processing and Computational Linguistics.* Berlin: Springer, pp. 82-95.

Ramasamy, L., and Žabokrtský, Z. 2012. Prague Dependency Style Treebank for Tamil. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

Rayson, P., Archer, D., Piao, S., and McEnery, A. M. 2004. *The UCREL Semantic Analysis System*. Retrieved on 21st January, 2018. http://eprints.lancs.ac.uk/1783/

Richard Socher, John Bauer, Christopher D. Manning and Andrew Y. Ng 2013. Parsing with Compositional Vector Grammars. In *ACL*. Vol 1, pp. 544-465.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell III, J. T., and Johnson, M. 2002. Parsing the Wall Street Journal Using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th ACL,* pp. 271-278.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of ACL*.

Sag, I. 1995. HPSG: Background and Basics, Stanford, CSLI; Retrieved on 1st November 2017. http: //hpsg.stanford.edu/hpsg.

Santorini, B. 1990. Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd revision). In *Technical Reports (CIS)*, pp. 570.

Seraji, M., Jahani, C., Megyesi, B., & Nivre, J. 2014. A Persian Treebank with Stanford Typed Dependencies. In *the 9th International Conference on Language Resources and Evaluation (LREC),* pp. 796-801.

Stanford Parser. Retrived on 1st June, 2018. https://nlp.stanford.edu/software/srparser.html.

Steedman, M. 1996. A Very Short Introduction to CCG. *Unpublished paper.* Retrieved on 21st September, 2018. http://www.coqsci.ed.ac.uk/steedman/paper.html

Steedman, M., and Baldridge, J. 2011. Combinatory Categorial Grammar. In *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, pp. 181-224.

Stolcke, A. 1995. An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities. In *Computational linguistics*, Vol.21. No.2, pp. 165-201.

Sureka, K., Srinivasagan, K. G., and Suganthi, S. 2014. An Efficiency Dependency Parser Using Hybrid Approach for Tamil Language. Retrieved on 20th December, 2017. https://arxiv.org/abs/1403.6381

Tammewar, A. 2015. *Cost Effective Dependency Parsing for Indian Languages.*Hyderabad: IIIT. MS dissertation.

Tandon, J., and Sharma, D. M. 2017. Unity in Diversity: A Unified Parsing Strategy for Major Indian Languages. In *Proceedings of the Fourth International Conference on Dependency Linguistics,* pp. 255-265.

Taylor, A., Marcus, M., and Santorini, B. 2003. The Penn Treebank: An Overview. In *Treebanks*. Dordrecht: Springer, pp. 5-22.

Titov, I., and Henderson, J. 2010. A Latent Variable Model for Generative Dependency Parsing. In *Trends in Parsing Technology.* Netherland: Springer, pp. 35-55.

UCREL Parsing Tagset. Retrieved on 1st June, 2018. http://ucrel.lancs.ac.uk.

Universal Dependency (UD) Tagset. Retrieved on 1st June, 2018. http://universaldependencies.org

H. Yamada and Y. Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pp. 195–206.

Towards Building a Dependency Parser for Tamil: A Discussion on Tags

Keerthana B. And K. Parameswari
tulips91@gmail.com and parameshkrishnaa@gmail.com

Abstract
This paper discusses about the available annotation guidelines for building a parser in Indian languages and does a detailed comparative study between AnnCorra and Universal Dependencies tagset as these two are the prominent tagsets used for building a parser for Tamil in the recent past. A statistical study of tags used and the importance of language specific tags are highlighted.

1. Introduction

Annotation guidelines are backbone in developing treebanks for parsers. These guidelines are built based on available grammar formalisms and are framed at various levels- morphological tags, POS tags and syntactic relation tags. The widely used annotation guidelines for Indian languages in the recent past are AnnCorra (Bharati, A., Sangal, R., Sharma, D. M., & Bai, L., 2006) and Universal Dependencies guidelines (Nivre, J., De Marneffe, M. C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., ... & Tsarfaty, R. 2016), which are built based on dependency grammars. A statistical and comparative study is done between these tagsets in this paper.

2. A survey on grammar formalisms

Grammar formalisms are essential in building the annotation guidelines as they define the linguistic properties. Some suitable grammar formalisms for building a parser includes:

2.1. Generalized Phrase Structure Grammar (GPSG)

It is a constraint-based grammar, deriving from constituency grammar, developed by Gerald Gazdar in the 1970s with Ewan Klein, Ivan Sag, and Geoffrey Pullum for English (Cf. Gazdar, G., et.al, 1985). Implemented languages include English, Persian, French, Chinese and Arabic (Bahrani, M. et.al., 2011). For example,

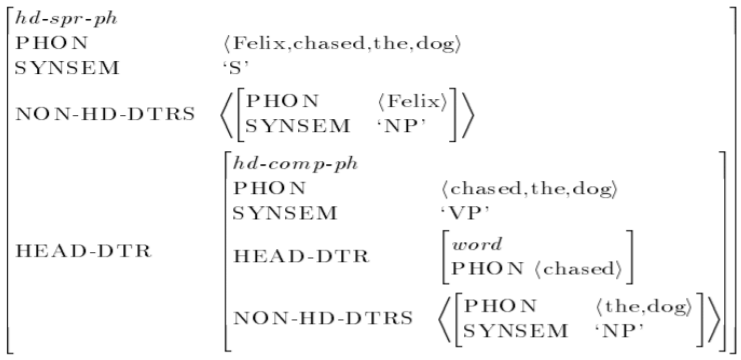
He gave him a book

((NP-he (N)))((VP-gave (V)))((NP- him (N)))((NP-a (DET) book (N)))

2.2. Head-driven Phrase Structure Grammar (HPSG)

It is lexical- based, constrained PSG, developed by Carl Pollard and Ivan Sag (Cf. Pollard, C., and Sag, I. A., 1994). Applicable languages include Romance languages, Slavic languages, German, Japanese, Welsh, English, Korean and Warlpiri (Levine, R. D. and Meurers, W. D., 2006). For example,

Felix chased the dog

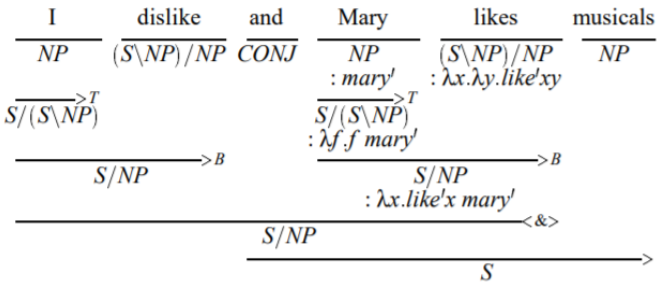


(extracted from Sag, I. A., 1995:15)

2.3. Combinatory Categorial Grammar

A lexicalised grammar form where categorial grammar is extended with functional operators, developed by Mark Steedman and Remo Pareschi (1987) and Szabolcsi (1992). It is applied in English (Hockenmaier, J., and Steedman, M., 2002, 342). For example,

I dislike and Mary likes musicals

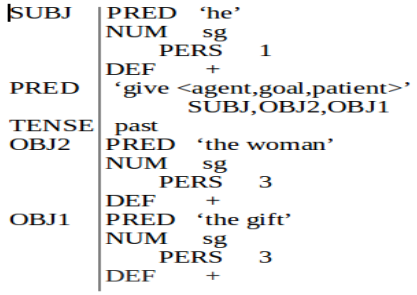


(extracted from Mark Steedman, 1996 (4))

2.4. Lexical-Functional Grammar

LFG is first published by Joan Bresnan (1982), represented in constituent and functional structure. It is used in parsing Wall Street Journal (WSJ) by Stefan Riezler, et.al. (2002). The major advantage found in LFG is that the mismatch between the surface structure and the deep argument structure as discussed in Chomskyan framework is not found here. For example,

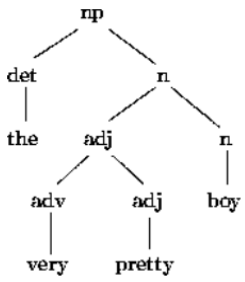
He gave the woman the gift



2.5. Tree Adjoining Grammar (TAG)

Tree Adjoining Grammar, formulated by Aravind Joshi (A. K. Joshi, Levy, and Takahashi, 1975) has both lexicalised and constraint-based variations. Elementary trees are combined here with substitution and adjunction operations (Kroch, A. S., & Joshi, A. K., 1985). It is applied in English and results obtained are better than previously mentioned formalisms (XTAG Research Group 1998; Abeille, A.; Bishop, K., Cote, Sharon, & Schabes, Y. 1990). For example,

The very pretty boy

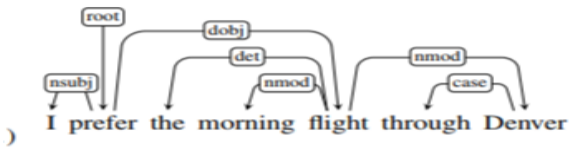


(extracted from Tree Adjoining Grammars, https://www.let.rug.nl/~vannoord/papers/diss/diss/node59.html)

2.6. Dependency Grammar (DG)

Dependency approach (both projective and non-projective) follows dependency grammar, tracing back to Panini's grammar. The modern thought of DG was proposed by a French linguist Lucien Tesnière during post-1950s. It represents the relation between the head and its dependents. Content words are marked by dependency relations; functional words attach to the content words they modify and punctuation attach to the head of the phrase/ clause. For example,

I prefer the morning flight through Denver.



(extracted from Speech and Language Processing (Jurafsky, D., & Martin, J. H., 2018))

For a morphologically rich and constituent-free language like Tamil, implementing

dependency model is better (Falavarjani, S. A. M., & Ghassem-Sani, G., 2015). Faster manual annotation and more efficient parsing is applicable for any language in DG (Jurafsky, D., & Martin, J. H., 2018).

3. Survey on available tagsets

A detailed survey is done on the major available tagsets, including AnnCorra tagset, Universal Dependencies tagset, Stanford dependencies tagset, Penn tagset, Prague tagset and Chinese Dependency tagset (Appendix 1). Among these, Universal Dependencies and AnnCorra tagsets are found to be implemented for Tamil following dependency grammar.

3.1. AnnCorra Tagset

Annotated Corpora (AnnCorra), a Pāninian Dependency grammar based tagset is built on the basis of *kāraka* and non-*kāraka* relations. It major goal is to have a uniform representation of annotated corpus of Indian languages (Bharati, A., et.al, 2002). It is initially built for parsing Hindi sentences and thus, the tags presented are according to Hindi grammar. Later, the same guidelines were adapted for other Indian languages (Marathi, Urdu, Bengali, Kannada, Telugu, Tamil, and Malayalam) (Cf. Tandon, J. and Sharma, D. M., 2017). It was even used by Amita, A. J. (2015) for English, in which HyDT annotation scheme and hybrid approach (statistical+ rule based) were used for parsing 2000 words.

There are 19 *kāraka* relations[5] and 25 Non-*kāraka* relations[6] existing in the tagset. The unique relations include jk1, mk1, k1s, k2g, k2p, k2s, k4a, k7t, k7p, and k7a. The following table represents the relations with examples:

S. No.	Tag	Examples
1	pk1	*e̲n̲̲n̲ai* ***avan̲*** *vēlai ceyvittān̲* 'He made me do the work'

[5] k1 (*karta* 'doer/agent/ subject'), pk1 (*prayojaka karta* 'causer'), jk1 (*prayojya karta* 'causee'), mk1 (*madhyastha karta*'mediator-causer'), k1s (*karta samanadhikarana-* 'noun complement of *karta*'), k2 (*karma* 'object/patient'), k2p (Goal, Destination), k2g (secondary *karma*), k2s (*karma samanadhikarana* 'object complement'), k3 (*karana* 'instrument'), k4 (*sampradana* 'recipient'), k4a (*anubhava karta* 'Experiencer'), k5 (*apadana* 'source'), k5prk (prakruti apadana 'source material'), k7t (*kAlAdhikarana* 'location in time'), k7p (*deshadhikarana* 'location in space'), k7 (*vishayadhikarana* 'location elsewhere'), k7a (according to) and k*u (*sAdrishya* 'similarity/comparison')

[6] r6 (*shashthi* 'genitive/possessive'), r6-k1, r6-k2 (*karta* or *karma* of a conjunct verb (complex predicate)), r6v (*kA* 'relation between a noun and a verb), adv (*kriyAvisheSaNa* 'manner adverbs'), sent-adv (Sentential Adverbs), rd (direction), rh (*hetu* 'reason'), rt (*tadarthya* 'purpose'), ras-k* (*upapada sahakArakatwa* 'associative'), ras-neg (Negation in Associative), rs (noun elaboration), rsp (address terms), nmod__relc, jjmod__relc, rbmod__relc (relative clauses, *jo-vo* constructions), nmod (participles etc. modifying nouns), vmod (verb modifier), jjmod (D-Rel modifiers of the adjectives), pof (part of units such as conjunct verbs), ccof (co-ordination and sub-ordination), fragof (Fragment of), enm (enumerator), rsym (ag for a symbol) and psp_cl (relation between clause and postposition following that clause)

2	jk1	***eṉṉai*** *avaṉ vēlai ceyvittāṉ* 'He made me do the work'
3	mk1	*eṉṉai avaṉ* ***ammāvaik koṇtu*** *vēlai ceyvittāṉ* 'He made mother to make me do the work'
4	k1s	***nāṉ*** *maruttuvar* 'I am doctor'
5	k2g	*nēruvai* ***māmā*** *eṉavum aẕaittaṉar* 'They also called Nehru as uncle'
6	k2p	*nāṉ* ***amērikkāviṙkuc*** *ceṉṙēṉ* 'I went to America'
7	k2s	*eṉṉai* ***putticāli*** *eṉak karutiṉar* 'They considered me as intelligent'
8	k4a	***eṉakku*** *kuḷirkiṉṙatu* 'I am feeling cold'
9	k7t	*iŋku* ***nēṙṙu*** *maẕai peytatu* 'It rained here yesterday'
10	k7p	*puttakam* ***paiyil*** *uḷḷatu* 'The book is in the bag'
11	k7a	*eṉ nāy* ***amērikkāvil*** *uḷḷatu* 'My dog is in America'

4.2. Universal Dependencies

Universal Dependency is a cross-linguistic project, built with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective (http://universaldependencies.org). The idea of annotation scheme has been taken from Stanford dependencies, Google universal part-of-speech tags and the Interset interlingua for morpho-syntactic tagsets in 2013 (McDonald et al., 2013). The parser has a lesser number of modules (Pre-processing (transliteration, sentence segmentation, tokenization); M-layer annotation (positional tagging) and A- layer annotation (dependency annotation), making it universal for any language typology. Indian languages like Hindi, Marathi, Sanskrit, Tamil, Telugu, and Urdu are included in the existing UD Treebank and Kannada and Pnar are upcoming languages listed in the UD website. There tagset is rich with 37 coarse grained tags[7]. Added to it, there are 198 language specific tags that are used in various languages

[7] *acl* (clausal modifier of noun (adjectival clause)), *advcl* (adverbial clause modifier), *advmod* (adverbial modifier), *amod* (adjectival modifier), *appos* (appositional modifier), *aux* (auxiliary), *case* (case marking), *cc* (coordinating conjunction), *ccomp* (clausal complement), *clf* (classifier), *compound* (compound), *conj* (conjunct), *cop* (copula), *csubj* (clausal subject), *dep* (unspecified dependency), *det* (determiner), *discourse* (discourse element), *dislocated* (dislocated elements), *expl* (expletive), *fixed* (fixed multiword expression), *flat* (flat multiword expression), *goeswith* (goes with), *iobj* (indirect object), *list* (list), *mark* (marker), *nmod* (nominal modifier), *nsubj* (nominal subject), *nummod* (numeric modifier), *obj* (object), *obl* (oblique nominal), *orphan* (orphan), *parataxis* (parataxis), *punct* (punctuation), *reparandum* (overridden disfluency), *root* (root), *vocative* (vocative) and *xcomp* (open clausal complement).

parsing system.

The uniqueness of this tagset lies in the inter-clausal tags, including *acl, advcl, ccomp, xcomp,* and *csubj.* The following table describes the richness of this tagset with examples:

S. No.	Tag	Examples
1	acl	***itaṉāl*** *avarkaḷ kaitu ceyyap paṭakkūṭum enak karutappaṭṭatu* 'It was thought that she might get arrested because of this'
2	advcl	*nāṉ* ***iruntāl*** *uāṉkku eṉṉa payam?* 'What is scary for you if I am there?'
3	ccomp	*uṉakkup paṭikkap* ***piṭikkum*** *eṉa avar coṉṉār* 'He said that you like to read'
4	xcomp	*avar* ***varaiya*** *ārampittār* 'He started to draw'
5	csubj	*avar* ***coṉṉatu*** *arttamuḷḷa oṉṟu* 'What he said is meaningful'

4. Comparative study between AnnCorra and Universal Dependencies tags

(i) AnnCorra tagset has a better representation of case which is essential for any morphologically rich language. A unique tag is given to each case and thus, a deep analysis is seen. For example, locative case has multiple functions and accordingly case is marked.

Tag: k7 (location elsewhere), k7t (location in time), k7p (location in space)

For instance:

(1) *iŋku* ***nēṟṟu*** *maẓai peytatu* 'It rained here yesterday' is marked k7t

(2) *puttakam* ***paiyil*** *uḷḷatu* 'The book is in the bag' is marked k7p

(3) *eṉ nāy* ***amērikkāvil*** *uḷḷatu* 'My dog is in America' is marked location elsewhere

The same is not found in the UPOS tag of UD. Instead, the differentiation is done in language specific tags. The tag 'obl' (oblique nominal) is generally used for all the non-core arguments of a clause. The distinction is done in language specific tags according to the language's requirement. (1) is marked obl:tmod (temporal modifier), (2) is marked as obl:arg (argument), (3) is marked as obl:loc (location). Among these, 'obl:arg' is already introduced in UD. The rest of the tags are available in other languages and similar occurrences are found in Tamil as well.

Similar cases are seen in other case markers as well.

(ii) UD has an inter-chunk and intra-chunk representation unlike AnnCorra. AnnCorra

has only intra-chunk tags. For instance,

Tag: acl in UD
(4) pa*ḷḷikku* ***celvataṟkāṉa*** *kāraṇam eṉṉa?* 'What is the reason for going to school?'
Here, in *celvataṟku+āṉa,* the former is marked 'acl' to the latter. This inter-clausal relation is absent in AnnCorra.

(iii) Dative subject constructions are marked in AnnCorra as k4a, whereas it is yet to be given a tag in UD for Tamil. The tag 'nsubj:nc' (non-canonical subjects) is marked in Telugu UD, which has to be extended for Tamil as well.

For instance,
(5) ***eṉakku*** *kuḷirkiṉṟatu* 'I am feeling cold' is an experiencer and not the real subject of the sentence.

5. Conclusion
The UD tags seem to be shallow with respect to the AnnCorra tags. But they are accommodated as language specific relations. The problem of inter-chunk tags seems to be unresolved in AnnCorra, which is an added plus point to UD system. Thus, Universal Dependencies has a wider scope in parsing Tamil sentences.

References

Bharati, A., Sangal, R., Sharma, D. M., & Bai, L. 2006. AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. *LTRC-TR31*, 1-38.

Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. A. 1985. *Generalized Phrase Structure Grammar*. Cambridge: Harvard University Press.

Nivre, J., De Marneffe, M. C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., ... & Tsarfaty, R. 2016. Universal dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016),* pp. 1659-1666.

Steedman, Mark and Remo Pareschi 1987. A Lazy way to Chart-Parse with Categorial Grammars. *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, Stanford CA, 81-88.

Falavarjani, S. A. M., & Ghassem-Sani, G. 2015. Advantages of Dependency Parsing for Free Word Order Natural Languages. In *International Conference on Current Trends in Theory and Practice of Informatics.* B**erlin: Springer,** pp. 511-518.

Jurafsky, D., & Martin, J. H. 2018. *Speech and Language Processing.* London: Pearson. Vol.3, pp 270. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

Joshi,A.K., L. S. Levy, & M. Takahashi 1975. Tree adjunct grammars. *Journal Computer Systems Science.* Vol.10.

Bahrani, M., Sameti, H., and Manshadi, M. H. 2011. *A Computational Grammar for Persian Based on GPSG*. Retrieved on 21st November, 2017. https://link.springer.com/article/10.1007/s10579-011-9144-1

Pollard, C., and Sag, I. A. 1994. *Head-Driven Phrase Structure Grammar*. US: University of Chicago Press.

Levine, R. D., and Meurers, W. D. 2006. Head-Driven Phrase Structure Grammar. In *Encyclopedia of Language and Linguistics*. Vol. 5, pp. 237-52.

Szabolcsi, Anna 1992. On Combinatory Grammar and Projection from the Lexicon. In Ivan Sag & Anna Szabolcsi (ed.), Lexical Matters. *CSLI, pp.* 241-268.

Hockenmaier, J., & Steedman, M. 2002. Generative Models for Statistical Parsing with Combinatory Categorial Grammar. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics,* pp. 335-342.

Bresnan, J. 1982. Control and complementation. In *Linguistic inquiry*. Vol. 13. No. 3, pp. 343-434.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell III, J. T., and Johnson, M. 2002. Parsing the Wall Street Journal Using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th ACL,* pp. 271-278.

Bharati, A., Sangal, R., Chaitanya, V., Kulkarni, A., Sharma, D. M., and Ramakrishnamacharyulu, K. V. 2002. AnnCorra: Building Tree-banks in Indian Languages. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization: Association for Computational Linguistics*. Vol. 12, pp. 1-8.

Universal Dependency (UD) Tagset. Retrieved on 1st June, 2018. https://universaldependencies.org

Taylor, A., Marcus, M., and Santorini, B. 2003. The Penn Treebank: An Overview. In *Treebanks*. Dordrecht: Springer, pp. 5-22.

Kroch, A. S., & Joshi, A. K. 1985. The Linguistic Relevance of Tree Adjoining Grammar.

Technical Reports (CIS), pp.671.

XTAG Research Group 1998. A Lexicalized Tree Adjoining Grammar for English. https://arxiv.org/abs/cs/9809024

Abeille, A., K.Bishop, Sharon Cote, and Y. Schabes 1990. *A Lexicalized Tree Adjoining Grammar for English*. Technical Report MS-CIS-90-24, Department of Computer and Information Science, University of Pennsylvania.

Tandon, J., and Sharma, D. M. 2017. Unity in Diversity: A Unified Parsing Strategy for Major Indian Languages. In *Proceedings of the Fourth International Conference on Dependency Linguistics,* pp. 255-265.

Amita, A. J. 2015. An Annotation Scheme for English Language Using Paninian Framework. *IJISET*, Vol. 2, pp. 616-619.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., ... and Bedini, C. 2013. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st ACL.* Vol. 2, pp. 92-97.

Appendix 1

S.No.	Category	AnnCorra Tagset	Universal Dependency Tagset		Stanford Tagset	Penn Tagset	Prague Tagset	Chinese Dependency Tagset
			GEN	TAM				
ADVERB								
	Adverbial Clause (Modifier)	sent-adv	advcl	advcl	advcl			
	Adverbial Modifier	adv (only manner verbs)	advmod	advmod	advmod	ADV		ADV
	Verb modifier	vmod						
	Adverbial modifier: emph			advmod:emph				
	Adverbial clause: relative clause	rbmod_relc	acl:relcl					
	Noun phrase as adverbial modifier		nmod:npmod	nmod:npmod	npadvmod			
	Reduced non-finite verbal modifier				vmod			
ARGUMENTS AND ADJUNCTS								
	a. COMPLEMENTS							
	clausal complement		ccomp	ccomp	ccomp		COMP	CMP
	Open clausal complement		xcomp	xcomp	xcomp			
	Object complement	k2s						
	Adjectival complement				acomp			
	Locative complements of put					PUT		
	b. ADJUNCTS							
	Consequence						CSQ	
	Effect						EFF	
	Result						RESL	
	Reason	Rh					REAS	
	Purpose	Rt						
	Manner						MANN	
	Left adjunct							LAD
	Right adjunct							RAD
NOMINAL MODIFIERS								
	Nominal modifier	nmod	nmod	nmod		NOM		
	Nominal modifier: possessive		nmod:poss					
	Nominal modifier: temporal (time)		nmod:tmod	nmod:tmod				
	Nominal modifier: position		nmod:in	nmod:in				
	Oblique nominal		obl	obl				
	Adjective clause		acl	acl				
	Adjectival clause: relative	jjmod_rel						

	clause	c						
ADJECTIVAL MODIFIERS								
	Adjectival modifier	jjmod	amod	amod	amod			
NEGATION								
	Negation modifier		neg		neg			
NUMBER								
	Numeric modifier	enm	nummod	nummod	num			
	Possession modifier				poss			
	Noun compound modifier				nn			
	Appositional Modifier		appos	appos	appos		APPS	
	Temporal modifier				tmod			
	Quantifier phrase modifier				quantmod			
ADPOSITION								
	Preposition-object				pobj			POB
	Prepositional complement				pcomp			
	Prepositional modifier				prep			
	Prepositional clausal modifier				prepc			
CASE								
	Case Marking		case	case				
	Locative (space)	k7p				LOC_TMP (temporal)	LOC	
	Location in time	k7t						
	Location elsewhere	k7				LOC		
	Location (manner)					LOC_MNR		
	Location (purpose/ reason)					LOC_PRP		
	Genetive/possessive	r6			poss			
	Associative	ras k*						
	Negation in associative	ras-NEG						
	Benefactive					BNF	BEN	
	Dative					DTV		
	Ethical Dative						ETHD	
	Instrument	k3						
	Vocative	Rad	vocative			VOC	VOC	
	Vocative in apposition						VOCAT	
SUBJECT								
	Clausal subject		csubj	csubj	csubj			
	Nominal/surface subject		nsubj	nsubj	nsubj	SBJ		SBV
	Clausal passive subject		csubj:pass		csubjpass			
	Passive nominal/ logical subject			nsubj:pass	nsubjpass	LGS		
	Actor/bearer/*karta*/doer/subject/agent	k1			agent		ACT	
	Noun complement of *karta*	k1s						
	Controlling subject				xsubj			
OBJECT								

	Direct object			dobj		dobj			
	Indirect object			iobj	iobj				IOB
	Fronting object								FOB
	Patient/object/*karma*		k2	obj	obj			PAT	VOB
OTHER THEMATIC ROLES									
	Cause	Causer	pk1					CAUS	
		Causee	jk1						
		Mediator-causer	mk1						
	Source		k5						
	Karma (Goal/destination)		k2p						
	Secondary karma		k2g						
	Source (with verbs denoting change of state)		k5prk						
	Directional (from)		Rd				DIR	DIR1	
	Directional (which way)							DIR2	
	Directional (where to)							DIR3	
AUXILIARY									
	Auxiliary			aux	aux	aux			
	Passive auxiliary			aux:pass		auxpass			
NUMBER									
	Element of compound number					number			
COMPOUNDS AND MULTI-WORD EXPRESSIONS									
	Compound		pof-compound	compound	compound				
	Compound for particle verb				compound:prt				
	Compound for serial verbs				compound:svc				
	Part of units- idiom		pof-idiom						
	Multi-word expression			mwe		mwe			
	Fixed multi word expression			fixed					
	Relation between noun and verb		r6v						
	Sentence type		Stype						
	Classifier			clf					
	Coordinating conjunction		ccof (for subordination as well)	cc	cc	cc			COO
	Fragment of		fragof						
	Expletive			expl		expl			
	Root			root	root	root		SENT	
	Conjunct		pof	conj	conj	conj		CONJ	
	Copula			cop	cop	cop			
	Dislocated elements			disocated					
	Unspecified dependency			dep					
	Determiner			det	det	det			

	Discourse element		discourse					
	Double roles: subject and object							DBL
	Foreign words		foreign					
	Goes with		goeswith					
	List		list					
	Marker		mark	mark	mark			
	Name		name					
	Orphan		orphan					
	Parataxis		parataxis	parataxis	Parataxis			
	Punctuation		punct	punct	punct			
	Overridden dis fluency		reparandum					
	same person's name (sur name)		flat/flat:name	flat/flat:name				
	Predicate					PRD	PRED	
	Denomination						DENOM	
	Sentence particles						PARTL	
	Empty verb						EV	
	Addressee						ADDR	
	Origin						ORIG	
	Accompaniment						ACMP	
	Aim						AIM	
	Attitude						ATT	
	Attributive							ATT
	Comparison						CPR	
	Concession						CNCS	
	Condition						COND	
	Confrontation						CONFR	
	Counterfactual						CTERF	
	Criterion						CRIT	
	Difference						DIFF	
	Part of phraseme						DPHR	
	Extent					EXT	EXT	
	Heritage						HER	
	Intensification						INTF	
	Intent						INTT	
	Means						MEANS	
	Adverb of modality						MOD	
	Norm						NORM	
	Reference to preceding text						PREC	
	Regard						REG	
	Rhematizer						RHE	

							M	
	Restriction						REST R	
	Substitution						SUBS	
	When	rsp					TWH EN	
	Since when						TSIN	
	Till when						TTIL L	
	How long						THL	
	For how long						TFHL	
	How often						THO	
	Parallel, contemporary						TPAR	
	From when						TFRW H	
	To when						TOW H	
	Appurtenance						APP	
	Descriptive						DES	
	Identity						ID	
	Material						MAT	
	Restrictive						RSTR	
	Disjunction						DISJ	
	Gradation						GRA D	
	Adversative						ADVS	
	Parenthesis						PAR	
	Similarity	k*u						
	Noun elaboration	rs						
	Independent structure							IS
	Head							HED
	Topicalized					TPC		
	Closely related					CLR		
	Cleft					CLF		
	Headline					HLN		
	Title					TTL		
	Dependent				dep			
	Discourse element				discour se			
	Pre-conjunct				preconj			
	Pre-determiner				predet			
	Phrasal verb particle				prt			
	Referent				ref			

Appendix

Sports data

```
# Sent_id = 1
#text= ஒரே நாளில் 16 விக்கெட்டுகள்...
# url =
https://sports.vikatan.com/cricket/16-wickets-in-a-single-day-first-day-match-report-of-ind
-vs-sl-pink-ball-test
1       ஒரே     ஒரே     DET     _       _       2       det     2:det   -
2       நாளில்  நாள்    NOUN    _       Case=Loc|Number=Sing 4       obl     4:obl:lmod      -
3       16      16      NUM     _       _       4       nummod  4:nummod        -
4       விக்கெட்டுகள்   விக்கெட்டு      NOUN    _       Case=Nom|Number=Plur 0       root    0:root
-
5       ...     ...     PUNCT   _       _       4       punct   4:punct-

# Sent_id = 2
#text= இன்றே முடிந்துவிடுமா பிங்க் பால் டெஸ்ட்?
# url =
https://sports.vikatan.com/cricket/16-wickets-in-a-single-day-first-day-match-report-of-ind
-vs-sl-pink-ball-test
1       இன்றே   இன்றே   NOUN    _       _       2       obl     2:obl   -
2-3     முடிந்துவிடுமா  முடிந்துவிடுமா  _       _       _       _       _       _       _
2       முடிந்து        முடி    VERB    _       Polarity=Pos|VerbForm=Conv  0       root    0:root
SpaceAfter=No
3       விடுமா  விடு    AUX     _       Gender=Neut|Number=Sing|Person=3    2       aux     2:aux
-
4       பிங்க்  பிங்க்  NOUN    _       Case=Nom|Number=Sing 6       compound        6:compound
-
5       பால்    பால்    NOUN    _       Case=Nom|Number=Sing 6       compound        6:compound
-
6       டெஸ்ட்          டெஸ்ட்          NOUN    _       Case=Nom|Number=Sing 2       nsubj   2:nsubj
-
7       ?       ?       PUNCT   _       _       6       punct   6:punct-

# Sent_id = 3
#text= முதல் ஓவரின் ஐந்தாவது பந்தை அகர்வாலுக்கு ்புல்லராக அவுட்சைட் ஆ்பில் வீசினார் லக்மல்.
# url =
https://sports.vikatan.com/cricket/16-wickets-in-a-single-day-first-day-match-report-of-ind
-vs-sl-pink-ball-test
1       முதல்   முதல்   NOUN    _       Case=Nom|Number=Sing 2       det     2:det   -
2       ஓவரின்          ஓவர்    NOUN    _       Case=Gen|Number=Sing 4       nmod:poss
4:nmod:poss     -
3       ஐந்தாவது       ஐந்தாவது       ADJ     _       _       4       amod    4:amod  -
4       பந்தை   பந்து   NOUN    _       Case=Acc|Number=Sing 9       obj     9:obj   -
5       அகர்வாலுக்கு    அகர்வால்       PROPN   _       Case=Dat|Number=Sing 9       obl     9:obl
-
6       ்புல்லராக       ்புல்லராக       ADV     _       _       9       advmod  9:advmod        -
7       அவுட்சைட்       அவுட்சைட்       NOUN    _       Case=Nom|Number=Sing 8       compound
8:compound      -
8       ஆ்பில்  ஆ்பில்  NOUN    _       Case=Nom|Number=Sing 9       obl     9:obl   -
9       வீசினார்        வீசு    VERB    _
Gender=Com|Number=Sing|Person=3|Polite=Form|Tense=Past|VerbForm=Fin    0       root    0:root
-
```

```
10      லக்மல் லக்மல் PROPN  _       Case=Nom|Number=Sing 9      nsubj  9:nsubj-
11      .       .       PUNCT  _       _       10     punct  10:punct      -

# Sent_id = 4
#text= பந்து பிட்சான இடத்தில் புழுதி நன்றாக எழும்பியது.
# url =
https://sports.vikatan.com/cricket/16-wickets-in-a-single-day-first-day-match-report-of-ind
-vs-sl-pink-ball-test
1       பந்து   பந்து   NOUN   _       Case=Nom|Number=Sing 6      nsubj  6:nsubj-
2       பிட்சான         பிட்சான         ADJ    _       _       3      amod   3:amod -
3       இடத்தில்        இடம்  NOUN   _       Case=Loc|Number=Sing 6      obl    6:obl:lmod
-
4       புழுதி  புழுதி  NOUN   _       Case=Nom|Number=Sing 6      nsubj  6:nsubj-
5       நன்றாக         நன்றாக         ADV    _       _       6      advmod 6:advmod      -
6       எழும்பியது     எழும்பு VERB   _
Gender=Neut|Number=Sing|Person=3|Polarity=Pos|Tense=Past|VerbForm=Part 0      root   0:root
-
7       .       .       PUNCT  _       _       6      punct  6:punct-

# Sent_id = 5
#text= அந்த ஒற்றை பந்தே ஆட்டத்தின் மொத்த போக்கையும் தெளிவாகக் கூறிவிட்டது.
# url =
https://sports.vikatan.com/cricket/16-wickets-in-a-single-day-first-day-match-report-of-ind
-vs-sl-pink-ball-test
1       அந்த    அந்த    DET    _       _       2      det    2:det  -
2       ஒற்றை ஒற்றை NOUN   _       Case=Nom|Number=Sing 10     nsubj  10:nsubj      -
3-4     பந்தே   பந்தே   _      _       _       _      _      _      _
3       பந்த்   பந்த்   NOUN   _       Case=Nom|Number=Sing 10     nsubj  10:nsubj
SpaceAfter=No
4       ஏ       ஏ       PART   _       _       3      mark   3:mark -
5       ஆட்டத்தின்      ஆட்டம்NOUN   _       Case=Gen|Number=Sing 7      nmod:poss
7:nmod:poss     -
6       மொத்த மொத்த DET    _       _       7      det    7:det  -
7-8     போக்கையும்      போக்கையும்      _      _       _      _      _      _      _
7       போக்கை  போக்கு NOUN   _       Case=Acc|Number=Sing 10     obj    10:obj
SpaceAfter=No
8       உம்     உம்     PART   _       _       7      mark   7:mark -
9       தெளிவாகக்      தெளிவாகக்      ADV    _       _       10     advmod 10:advmod     -
10      கூறி    கூறு    VERB   _       Polarity=Pos|VerbForm=Conv  0      root   0:root -
11      விட்டது         விடு   AUX    _
Gender=Neut|Number=Sing|Person=3|Tense=Past|VerbForm=Fin 10     aux    10:aux -
12      .       .       PUNCT  _       _       10     punct  10:punct      -
```

Agriculture Data

```
# sent_id = 1
# text = புல் இன வகைகளின் விதை ஆகும்.
1       புல்    புல்    NOUN   NNN-3SN--      Case=Nom|Gender=Neut|Number=Sing|Person=3  2
compound        _       TokenRange=158:162
2       இன      இனம்   PART   JJ-------      _       3      nmod   _       TokenRange=163:165
3       வகைகளின்       வகை   NOUN   NNG-3PN--      Case=Gen|Gender=Neut|Number=Plur|Person=3
33      nmod:poss       _       TokenRange=166:174
4       விதை விதை NOUN   NNN-3SN--      Case=Nom|Gender=Neut|Number=Sing|Person=3  0
root    _       TokenRange=175:179
```

```
5	ஆகும்	ஆகு	VERB	Vr-F3SNAA
Gender=Neut|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Fut|VerbForm=Fin|Voice=Act
4	cop	_	SpaceAfter=No|TokenRange=180:185
6	.	.	PUNCT	Z#-------	PunctType=Peri	4	punct	_
SpacesAfter=\r\n|TokenRange=185:186


# sent_id = 2
# text = இது தென்கிழக்காசியாவில் தோன்றியது.
1	இது	இது	PRON	RpN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3|PronType=Prs	3	nsubj	_
TokenRange=188:191
2	தென்கிழக்காசியாவில்	தென்கிழக்காசியா	PROPN	NEL-3SN--
Case=Loc|Gender=Neut|Number=Sing|Person=3	3	obl:loc	_	TokenRange=192:211
3	தோன்றியது	தோன்று	VERB	Vr-D3SNAA
Gender=Neut|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Past|VerbForm=Fin|Voice=Act
0	root	_	SpaceAfter=No|TokenRange=212:221
4	.	.	PUNCT	Z#-------	PunctType=Peri	3	punct	_
SpacesAfter=\r\n|TokenRange=221:222

# sent_id = 3
# text = இது ஈரநிலங்களில் வளரக்கூடியது.
1	இது	இது	PRON	RpN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3|PronType=Prs	3	nsubj	_
TokenRange=224:227
2	ஈரநிலங்களில்	ஈரநிலம்	NOUN	NNL-3PN--
Case=Loc|Gender=Neut|Number=Plur|Person=3	3	obl:loc	_	TokenRange=228:240
3	வளரக்கூடியது	வளரக்கூடு	VERB	Vr-D3SNAA
Gender=Neut|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Past|VerbForm=Fin|Voice=Act
0	root	_	SpaceAfter=No|TokenRange=241:253
4	.	.	PUNCT	Z#-------	PunctType=Peri	3	punct	_
SpacesAfter=\r\n|TokenRange=253:254

# sent_id = 4
# text = ஆனால், அரிசிக்கு முளைக்கும் திறன் கிடையாது.
1	ஆனால்	ஆனால்	ADV	AA-------	_	6	advmod	_
SpaceAfter=No|TokenRange=491:496
2	,	,	PUNCT	Z:-------	PunctType=Comm	3	punct	_
TokenRange=496:497
3	அரிசிக்கு	அரிசி	NOUN	NND-3SN--	Case=Dat|Gender=Neut|Number=Sing|Person=3
4	nsubj:nc	_	TokenRange=498:507
4	முளைக்கும்	முளை	VERB	Jd-F----A	Polarity=Pos|Tense=Fut|VerbForm=Part
5	acl	_	TokenRange=508:518
5	திறன்	திறன்	NOUN	NNN-3SN--	Case=Nom|Gender=Neut|Number=Sing|Person=3	6
obj	_	TokenRange=519:524
6	கிடையாது	கிடை	VERB	Vr-T3PNAA
Gender=Neut|Mood=Ind|Number=Plur|Person=3|Polarity=Pos|VerbForm=Fin|Voice=Act	0	root
_	SpaceAfter=No|TokenRange=525:533
7	.	.	PUNCT	Z#-------	PunctType=Peri	6	punct	_
SpacesAfter=\r\n|TokenRange=533:534
```

Tourism data

```
#Sent_id = 1	_	_	_
#text= புனித நீர்	_	-	-
```

```
#url = https://www.tamilnadutourism.tn.gov.in/tamil/destinations/agni-theertham-beach
15-03-2023 14:24:04                                     _               -       -
1       புனித   புனிதம் NOUN    _       Number=Sing     2       compound
2       நீர்    நீர்    NOUN    _       Case=Nom|Number=Sing    0       root

#Sent_id = 2                                    _               -       -
#text= இந்த நீர்நிலையில் ஹேங்கவுட் செய்வதற்கு ஒரு நிதானமான இடமாக இருப்பதை விட
அதிகமான விஷயங்கள் உள்ளன.                                     _               -       -
#url = https://www.tamilnadutourism.tn.gov.in/tamil/destinations/agni-theertham-beach
15-03-2023 14:24:04                                     _               -       -
1       இந்த    இந்த    DET     _       ###None 2       det
2       நீர்நிலையில்    நீர்நிலை        NOUN    _       Case=Loc|Number=Sing    4       obl_loc
3       ஹேங்கவுட்       ஹேங்கவுட்       NOUN    _       Case=Nom|Number=Sing    4       obj
4       செய்வதற்கு      செய்வது        NOUN    _       Case=Dat|Number=Sing    8       advcl
5       ஒரு     ஒன்று  DET     _       ###None 6       det
6       நிதானமான       நிதானமான       ADJ     _       ###None 7       amod
7       இடமாக  இடம்    ADV     _       Number=Sing     8       advmod
8       இருப்பதை        இருப்பது        NOUN    _       Case=Acc|Number=Sing    12      obl
9       விட     விட     ADP     _       ###None 8       case
10      அதிகமான        அதிகமான        ADJ     _       ###None 11      amod
11      விஷயங்கள்       விஷயம்  NOUN    _       Case=Nom|Number=Plur    12      nsubj
12      உள்ளன உள்      VERB    _       Gender=Neut|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin
0       root
13      .       .       PUNCT   _       ###None 12      punct

#Sent_id = 3                                    _               -       -
#text= இது ஒரு புனிதமான இடமாகும்.                                     _               -       -
#url = https://www.tamilnadutourism.tn.gov.in/tamil/destinations/agni-theertham-beach
15-03-2023 14:24:04                                     _               -       -
1       இது     இது    PRON    _       Case=Nom|Gender=Neut|Number=Sing|Person=3    4       nsubj
2       ஒரு     ஒன்று  DET     _       ###None 3       det
3       புனிதமான        புனிதமான       ADJ     _       ###None 4       amod
4-5     இடமாகும்        _       _       _       _       _       _
4       இடம்    இடம்    NOUN    _       Case=Nom|Number=Sing    0       root
5       ஆகும்   ஆகு     AUX     _       Gender=Neut|Number=Sing|Person=3|Tense=Fut|VerbForm=Fin
4       cop
6       .       .       PUNCT   _       ###None 5       punct

#Sent_id = 4                                    _               -       -
#text= இது ஆண்டு முழுவதும் பக்தர்களால் நிரம்பியுள்ளது.                                     _
-       -
#url = https://www.tamilnadutourism.tn.gov.in/tamil/destinations/agni-theertham-beach
15-03-2023 14:24:04                                     _               -       -
1       இது     இது    PRON    _       Case=Nom|Gender=Neut|Number=Sing|Person=3    5       nsubj
2       ஆண்டு   ஆண்டு  NOUN    _       Case=Nom|Number=Sing    5       obl:tmod
3       முழுவதும்       முழுவதும்       DET     _       ###None 2       det
4       பக்தர்களால்     பக்தர்  NOUN    _       Case=Ins|Number=Plur    5       obl:inst
5-6     நிரம்பியுள்ளது  _       _       _       _       _       _
5       நிரம்பி நிரம்பு VERB    _       Polarity=Pos|VerbForm=Conv      0       root
6       உள்ளது உள்     AUX     _       Gender=Neut|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
5       aux
7       .       .       PUNCT   _       ###None 6       punct

#Sent_id = 5                                    _               -       -
#text= இரட்சிப்பு மற்றும் கேளிக்கைகளை நாடுகிறது.
#url = https://www.tamilnadutourism.tn.gov.in/tamil/destinations/agni-theertham-beach
15-03-2023 14:24:04                                     _               -       -
1       இரட்சிப்பு      இரட்சிப்பு      NOUN    _       Case=Nom|Number=Sing    4       obj
2       மற்றும் மற்றும் CCONJ   _       ###None 3       cc
3       கேளிக்கைகளை     கேளிக்கை        NOUN    _       Case=Acc|Number=Plur    1       conj
4       நாடுகிறது       நாடு    VERB    _
Gender=Neut|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin    0       root
```

```
5       .       .       PUNCT   _       ###None 4       punct
```

Speech Conversation

```
# sent_id = 1
# text = அதுல ஒரு ம்யூசிலேஜ் வந்து வரும்.
1       அதுல    அதுல    PRON    RpN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3|PronType=Prs   4       obl     _
TokenRange=285:289
2       ஒரு     ஒரு     ADJ     JJ-------       _       3       amod    _       TokenRange=290:293
3       ம்யூசிலேஜ்      ம்யூசிலேஜ்      PROPN   NEN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3  4       nsubj   _       TokenRange=294:304
4       வந்து   வா      VERB    Vt-T---AA       Polarity=Pos|VerbForm=Part|Voice=Act       5
dep     _       TokenRange=305:310
5       வரும்   வரு     AUX     Vr-F3SNAA
Gender=Neut|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Fut|VerbForm=Fin|Voice=Act
0       root    _       SpaceAfter=No|TokenRange=311:316
6       .       .       PUNCT   Z#-------       PunctType=Peri4       punct   _
SpacesAfter=\r\n|TokenRange=316:317

# sent_id = 2
# text = சவச்சு சாப்பிடும்போது ம்யூசிலேஜ் இருக்கும்.
1       சவச்சு சவச்சு VERBNNN-3SN-- Case=Nom|Gender=Neut|Number=Sing|Person=3  2       advcl
_       TokenRange=319:325
2       சாப்பிடும்போது சாப்பிடும்போது VERB    AA-------       _       4       advcl   _
TokenRange=326:340
3       ம்யூசிலேஜ்      ம்யூசிலேஜ்      PROPN   NEN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3  4       nsubj   _       TokenRange=341:351
4       இருக்கும்       இரு     VERB    Vr-F3SNAA
Gender=Neut|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Fut|VerbForm=Fin|Voice=Act
0       root    _       SpaceAfter=No|TokenRange=352:361
5       .       .       PUNCT   Z#-------       PunctType=Peri4       punct   _
SpacesAfter=\r\n|TokenRange=361:362

# sent_id = 3
# text = ரொம்ப பாடி ஹீட் இருக்கு, இல்ல வைட் டிஸ்சார்ஜ் இருக்கு.
1       ரொம்ப ரொம்ப ADJ     NEN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  4
nmod    _       TokenRange=533:538
2       பாடி    பாடி    NOUN    NEN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  3
compound        _       TokenRange=539:543
3       ஹீட்    ஹீட்    NOUN    NEN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  4
nsubj   _       TokenRange=544:548
4       இருக்கு இரு     NOUN    NND-3SN--       Case=Dat|Gender=Neut|Number=Sing|Person=3  0
root    _       SpaceAfter=No|TokenRange=549:556
5       ,       ,       PUNCT   Z:-------       PunctType=Comm9       punct   _
TokenRange=556:557
6       இல்ல    இல்     SCONJ   NEN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  9
advmod  _       TokenRange=558:562
7       வைட்    வைட்    PROPN   NEN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  8
compound        _       TokenRange=563:567
8       டிஸ்சார்ஜ்      டிஸ்சார்ஜ்      PROPN   NEN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3  9       nsubj   _       TokenRange=568:578
9       இருக்கு இரு     VERB    Vr-P3SNAA       Case=Dat|Gender=Neut|Number=Sing|Person=3  4
conj    _       SpaceAfter=No|TokenRange=579:586
```

```
10      .       .       PUNCT   Z#-------       PunctType=Peri4       punct   _
SpacesAfter=\r\n|TokenRange=586:587

# sent_id = 4
# text = இல்ல, எனக்கு ரத்த கொழுப்ப குறைக்கணும், ரத்த கொழுப்பு சேர கூடாது.
1       இல்ல    இல்     ADV     AA-------       _       10      advmod  _
SpaceAfter=No|TokenRange=94:98
2       ,       ,       PUNCT   Z:-------       PunctType=Comm3       punct   _
TokenRange=98:99
3       எனக்கு  என்     PRON    RpD-1SA--
Animacy=Anim|Case=Dat|Gender=Com|Number=Sing|Person=1|PronType=Prs      6       nsubj:nc
_       TokenRange=100:106
4       ரத்த    ரத்த    NOUN    NO--3SN--       Gender=Neut|Number=Sing|Person=3    5
compound        _       TokenRange=107:111
5       கொழுப்ப        கொழுப்பு        NOUN    Vu-T---AA       Polarity=Pos|VerbForm=Inf|Voice=Act
6       obj     _       TokenRange=112:119
6       குறைக்கணும்    குறைக  VERB    Vr-F3SNAA
Gender=Neut|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Fut|VerbForm=Fin|Voice=Act
0       root    _       SpaceAfter=No|TokenRange=120:131
7       ,       ,       PUNCT   Z:-------       PunctType=Comm10      punct   _
TokenRange=131:132
8       ரத்த    ரத்த    NOUN    JJ-------       _       9       compound        _
TokenRange=133:137
9       கொழுப்பு        கொழுப்பு        NOUN    NNN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3  10      nsubj   _       TokenRange=138:146
10      சேர     சேர்    VERB    Vu-T---AA       Polarity=Pos|VerbForm=Inf|Voice=Act 6       conj
_       TokenRange=147:150
11      கூடாது  கூடு    AUX     VR-T3SN-N
Gender=Neut|Mood=Ind|Number=Sing|Person=3|Polarity=Neg|VerbForm=Fin     10      aux     _
SpaceAfter=No|TokenRange=151:157
12      .       .       PUNCT   Z#-------       PunctType=Peri10      punct   _
SpacesAfter=\r\n|TokenRange=157:158

# sent_id = 5
# text = அப்படி இருக்கிறவங்க ஊற வெச்சு எடுக்கலாம்.
1       அப்படி  அப்படி  ADV     AA-------       _       2       advmod  _       TokenRange=644:650
2       இருக்கிறவங்க   இரு     NOUN    VzNF3SNAA
Case=Nom|Gender=Neut|Number=Sing|Person=3|Polarity=Pos|Tense=Fut|VerbForm=Ger|Voice=Act
3       acl     _       TokenRange=651:663
3       ஊற     ஊற     VERB    Jd-D----A       _       4       xcomp   _       TokenRange=664:666
4       வெச்சு  வெச்சு  VERB    NNN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  5
advcl   _       TokenRange=667:673
5       எடுக்கலாம்      VERB    Vu-T---AA       Polarity=Pos|VerbForm=Inf|Voice=Act 0       root
_       _       SpaceAfter=No|TokenRange=674:684
6       .       .       PUNCT   Z#-------       PunctType=Peri5       punct   _
SpacesAfter=\r\n\r\n\r\n\r\n\s\s\r\n\r\n\r\n|TokenRange=684:685
```

Social media

```
# sent_id = 1
# text = நேற்று இரவு காட்சி #palazzo இல் பார்த்தேன்.
1       நேற்று  நேற்று  NOUN    AA-------       _       2       nmod    _       TokenRange=0:6
2       இரவு    இரவு    NOUN    NNN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  3
compound        _       TokenRange=7:11
3       காட்சி  காட்சி  NOUN    NNN-3SN--       Case=Nom|Gender=Neut|Number=Sing|Person=3  6
nsubj   _       TokenRange=12:18
```

```
4	#palazzo	#palazzo	PROPN	NNN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3	6	obl	_
SpacesAfter=\s\s|TokenRange=19:27
5	இல்	இல்	ADP	PP-------	AdpType=Post	4	case	_
TokenRange=29:32
6	பார்த்தேன்	பார்	VERB	Vr-T1SAAA
Animacy=Anim|Gender=Com|Mood=Ind|Number=Sing|Person=1|Polarity=Pos|VerbForm=Fin|Voice=Act
0	root	_	SpaceAfter=No|TokenRange=33:43
7	.	.	PUNCT	Z#-------	PunctType=Peri	6	punct	_
SpacesAfter=\s\r\n|TokenRange=43:44

# sent_id = 2
# text = 50 to 60 audience இருந்தார்கள்.
1	50	50	NUM	U=-------	NumForm=Digit|NumType=Card	4	nummod	_
TokenRange=47:49
2	to	to	ADP	Ux-------	NumType=Card	1	case	_
TokenRange=50:52
3	60	60	NUM	U=-------	NumForm=Digit|NumType=Card	4	nummod	_
TokenRange=53:55
4	audience	audience	NOUN	Ux-------
Animacy=Anim|Case=Nom|Gender=Com|Number=Plur|Person=3	5	nsubj	_
TokenRange=56:64
5	இருந்தார்கள்	இரு	VERB	NNN-3PA--
Animacy=Anim|Case=Nom|Gender=Com|Number=Plur|Person=3	0	root	_
SpaceAfter=No|TokenRange=65:77
6	.	.	PUNCT	Z#-------	PunctType=Peri	5	punct	_
SpacesAfter=\s\r\n|TokenRange=77:78

# sent_id = 3
# text = நல்ல கதாபாத்திர தேர்வு.
1	நல்ல	நல்ல	ADJ	JJ-------	_	2	amod	_	TokenRange=153:157
2	கதாபாத்திர	கதாபாத்திர	PROPN	NO--3SN--	Gender=Neut|Number=Sing|Person=3
3	nmod	TokenRange=158:168
3	தேர்வு	தேர்வு	NOUN	NNN-3SN--	Case=Nom|Gender=Neut|Number=Sing|Person=3	0
root	_	SpaceAfter=No|TokenRange=169:175
4	.	.	PUNCT	Z#-------	PunctType=Peri	3	punct	_
SpacesAfter=\s\r\n|TokenRange=175:176

# sent_id = 4
# text = மயில்சாமியின் நடிப்பு அட்டகாசம்.
1	மயில்சாமியின்	மயில்சாமி	PROPN	NEG-3SN--
Case=Gen|Gender=Neut|Number=Sing|Person=3	2	nmod:poss	_	TokenRange=228:241
2	நடிப்பு	நடிப்பு	NOUN	NNN-3SN--	Case=Nom|Gender=Neut|Number=Sing|Person=3	3
nsubj	_	TokenRange=242:249
3	அட்டகாசம்	அட்டகாசம்	NOUN	NNN-3SN--
Case=Nom|Gender=Neut|Number=Sing|Person=3	0	root	_
SpaceAfter=No|TokenRange=250:259
4	.	.	PUNCT	Z#-------	PunctType=Peri	3	punct	_
SpacesAfter=\s\r\n|TokenRange=259:260

# sent_id = 5
# text = Technically குட்.
1	Technically	technically	ADV	NEN-3SH--
Case=Nom|Gender=Neut|Number=Sing|Person=3	2	advmod	_	TokenRange=263:274
2	குட்	குட்	NOUN	NNN-3SN--	Case=Nom|Gender=Neut|Number=Sing|Person=3	0
root	_	SpaceAfter=No|TokenRange=275:279
3	.	.	PUNCT	Z#-------	PunctType=Peri	2	punct	_
SpacesAfter=\s\r\n|TokenRange=279:280
```

Domain Adaptation of Tamil Syntactic Parser: A Data-driven Approach

by Keerthana B

Submission date: 12-Mar-2024 10:48AM (UTC+0530)
Submission ID: 2318371416
File name: Copy_of_Final_Thesis_2_1.pdf (2.29M)
Word count: 24926
Character count: 129784

Domain Adaptation of Tamil Syntactic Parser: A Data-driven Approach

ORIGINALITY REPORT

2%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	universaldependencies.org Internet Source	1%
2	Submitted to University of Hyderabad, Hyderabad Student Paper	1%
3	www.uttamam.org Internet Source	<1%
4	aclanthology.org Internet Source	<1%
5	api.repository.cam.ac.uk Internet Source	<1%
6	pdfs.semanticscholar.org Internet Source	<1%
7	skemman.is Internet Source	<1%

Exclude quotes On

Exclude matches < 14 words

Exclude bibliography On