
A Documentation of Tamil Syntactic Parser

Project Name: A Syntactic Parser for Tamil: A Data-driven Approach

Funding Agency: **Tamil Virtual Academy (TVA)**

Letter No :Lr.N0.TVA/TC-EOI /2022/006-5, Dated. 05.01.2023

Report Submitted by,

Dr. Parameswari Krishnamurthy (PI of the Project)

Language Technology Research Center(LTRC),
International Institute of Information Technology, Hyderabad
Prof. C R Rao Road,
Gachibowli, Hyderabad 500 032,
Telangana, INDIA, Phone: +91 85000 25207, email: param.krishna@iiit.ac.in



27th February 2024

TABLE OF CONTENTS

TABLE OF CONTENTS.....	1
1. OVERVIEW OF THE PROJECT.....	3
2. MILESTONES.....	3
3. GUIDELINE PREPARATION:.....	4
3.1. POS tags: Guidelines.....	4
3.2 Morph features Guidelines.....	5
3.3 Multi-Token-word Split Guidelines.....	5
3.4 Treebank: Guidelines.....	5
3.5 Challenges.....	5
4. TAMIL TREEBANK.....	6
4.1 Data.....	6
4.1.1 Corpus Collection.....	6
4.2 Pre-processing.....	9
4.2.1 Corpus Cleaning.....	9
Noise cleaning.....	9
Unicode Normalization.....	9
Spelling Normalization.....	9
4.2.2 Tokenization.....	10
4.3 Annotation Process:.....	13
4.3.1 CONLL-U Format.....	13
4.3.2 Syntactic Tagset.....	15
4.4. Annotation Tool:.....	15
4.5. Statistics:.....	16
5. MODEL AND EVALUATION.....	20
5.1. Machine Learning Model.....	20
5.2 Evaluation.....	21

6. TAMIL PARSER GUI AND BACKEND.....	21
6.1 Introduction:.....	21
6.2 Overview:.....	21
6.3 Backend Model:.....	22
6.4 Client Side user interface.....	22
6.4.1 Parser:.....	22
6.4.2 Graph view:.....	22
6.4.3 Guidelines:.....	23
6.5 User Interface screenshots:.....	23
6.5.1 Tamil Syntactic Parser screenshot.....	23
6.5.2 Guidelines screenshot.....	24
6.5.3 Graph view screenshot.....	24
7. CONCLUSION.....	25
REFERENCES.....	25
FUND BUDGET BREAK-UP AFTER GST:.....	26

1. OVERVIEW OF THE PROJECT

This project aims to develop a syntactic parser for Tamil which can produce parse trees of an input sentence. A parser is an automated tool that provides a syntactic or syntactico-semantic analysis/representation of the relations of words in a sentence. The primary focus of this project is to build an automatic Tamil syntactic parser which provides the syntactic tree as a tool for further analysis. This project is both Research and Product-oriented with the following objectives.

- ☑ Compiling large corpora suitable for parsing Tamil
- ☑ Cleaning and normalising the corpora
- ☑ Build guidelines for pre-processing and annotating the cleaned corpora for Tamil
- ☑ Pre-process and linguistically annotate the text with NLP tools such as tokenizer, multi-word token segmenter, Parts-of-Speech (POS), and Morphological analyser
- ☑ Annotate the corpora with syntactic tags using universal dependency guidelines
- ☑ Build a machine learning model for parsing Tamil sentences
- ☑ Design a GUI for viewing Tamil parsed output as a syntactic tree.
- ☑ Evaluation of parser and further improvement

The selected objectives of the project are executed as a part of milestone 1 , milestone 2, milestone 3 and milestone 4

2. MILESTONES

Milestone-1:

Deliverables: Cleaned Modern Standard written Tami Corpora (100K tokens) suitable for parsing Tamil (Raw data) shall be given.

Period of Completion: Milestone activity 1 should be completed in 2 months from the date of acceptance of the work order.

Milestone-2:

Deliverables: Guidelines for annotation shall be given in the form of rules.

Period of Completion: Milestone activity 2 should be completed in 4 months from the date of completion of milestone activity.

Milestone-3:

Deliverables: Tokenized Annotated Corpora (100K tokens), which includes tokenization, multi-word token identification, Parts-of-speech (POS) tagging, morphologically analysed database along with syntactic tags shall be given.

Period of Completion: Milestone activity 3 should be completed in 5 months from the date of completion of milestone activity 2.

Milestone 4:

Deliverables: Syntactic Parsing tool and Documentation shall be given.

Period of Completion: Milestone activity 4 should be completed in 1 month from the date of completion of milestone activity 3.

3. GUIDELINE PREPARATION:

Guidelines for POS tagging, morphological feature identification, Syntactic tagging and multi-word token identification are prepared and we started to improve them during the course of annotation.

3.1. POS tags: Guidelines

[Guidelines for Parts of Speech Tagging](#)¹

¹ <https://docs.google.com/document/d/1YhBdH7TE5dxhkfXtxyskwoLun2ZJxUd4RGBYq81ZCbs/edit>

3.2 Morph features Guidelines

[Guidelines for Morph](#)

3.3 Multi-Token-word Split Guidelines

[Guidelines for Multi Token Word split](#)

3.4 Treebank: Guidelines

[Guidelines for Treebank](#)

3.5 Challenges

Annotating Tamil texts with the following tags can be challenging due to several reasons:

(i) Copula construction

Identifying copulas and their roles in linking subjects and predicates requires a deep understanding of Tamil syntax and semantics.

ஆகும் is connected to the predicate noun as copula with the POS as AUX

ஹீமோகுளோபின் என்பது இரத்தத்தில் உள்ள சிவப்பணுக்களின் அளவையாகும்.

Decision:

- (i) The predicate noun is the “root” of the sentence
- (ii) ஆகும் as copula, connected to the predicate

(ii) Multi-token words

Recognizing and properly segmenting multi-token words requires careful attention to morphology and word boundaries in Tamil.

Decision: Syntactically two different word forms which are written together is split.

Example: அளவையாகும் is split into அளவை + ஆகும்

(iii) Conjunction : Head initial Approach

Analyzing the positioning of heads and constituents demands a thorough grasp of Tamil syntactic structures.

அவனும் அவளும் வந்தார்கள் .

Here , அவனும் அவளும் are conjoined with conjunction marker -உம்

(iv) Compounds

Compounds: Identifying compound words and understanding their compositional meanings necessitates linguistic expertise and context awareness.

(v) xcomp vs advcl

xcomp vs advcl: Differentiating between types of clausal complements and adverbial clause modifiers demands precise syntactic analysis in Tamil sentences.

(vi)mark vs case

Mark vs case: Distinguishing between markers and cases in Tamil requires detailed knowledge of the language's grammatical features and nuances.

(vii) cc and cconj

CC and cconj: Recognizing coordinating conjunctions and their usage within compounds poses challenges, especially in complex sentence structures.

(viii) errors in the input sentences

Errors in the input sentences: Detecting and rectifying errors in the input text demands linguistic proficiency and careful attention to detail.

Overall, annotating Tamil texts with these tags requires a combination of linguistic expertise, cultural understanding, and attention to linguistic diversity within the Tamil language.

4. TAMIL TREEBANK

4.1 Data

4.1.1 Corpus Collection

1. Tourism
2. Health

3. Sports
4. Agriculture
5. Short Stories and Blogs

S. No	Domain	Website	Size		Copyright
			Tokens	Sentences	
1	Tourism	https://www.tamilnadutourism.tn.gov.in/tamil/destinations	127533	14609	Tamil Nadu Tourism (Government of Tamilnadu)
		TVA Coprus			TVA
		https://ta.wikipedia.org/wiki/விளையாட்டு			CC_BY_SA
2	Health	https://www.vikatan.com/health https://www.femina.in/tamil/health	459190	47804	Vikatan (Emailed.) Times of India (Emailed.)
3	Sports	https://ta.wikipedia.org/wiki/தமிழ்நாட்டில் விளையாட்டுக்கள் https://sports.vikatan.com/cricket ??5k	276478	30003	CC_BY_SA Vikatan (Emailed.)
4	Blogs and Short Stories	https://www.jeyamohan.in/ https://www.jeyamohan.in/சிறுகதைகள்-2	222621	34628	JeyaMohan (Emailed and got Permission)

5	Agriculture	https://tamil.oneindia.com/agriculture/ & - TVA Corpus	246142	32465	Greynium Information Technologies Pvt Ltd. (Emailed) & TVA
	Total		1331964	159509	

Table 1: Corpus Collection in Tamil

Type Token Ratio (TTR) , Lexical and its structural richness are taken as a criteria for selecting the sentences for annotation. Table 2 showing number of tokens selected from each domain and their TTR and sentences.

Domain	Tokens	Types	TTR	Sentences
Tourism	20394	8843	43.36	2153
Health	20009	9348	46.71	2054
Sports	20498	9745	47.54	2343
Agriculture	20601	10186	49.44	2467
Short stories and Blogs	20377	11530	56.58	3034
Total based on domains	1,01,879	49,652	48.72	12051

Table 2: Corpus Selection for Tamil Parsing

Type-Token Ratio (TTR)

The domain of corpus selected for Tamil parsing are combined and their TTR is calculated. This corpus is available here:

https://drive.google.com/file/d/1jQAutsr4PdNJ1pJXawP_7RHhOZfjZe33/view?usp=sharing

Tokens	Types	Sentences	TTR
1,01,879	38,594	12051	37.88

Table 3: Combined Corpus details selected for Tamil Parsing

TTR in Table 3 shows that the selected corpus has a good coverage of wordforms with unique tokens i.e. types which is important for our parsing task.

4.2 Pre-processing

4.2.1 Corpus Cleaning

Once the corpus is collected from various sources as given in Table 1, we have cleaned the corpus to make it useful for NLP tasks.

Noise cleaning

While crawling the corpus from the web pages it contains lot of noise and unnecessary information not suitable for parsing. Noise may include menu items, website name, external hyperlinks, advertisements, repetitive information like website header, footer, sidebar etc. All such information is excluded and made sure that only the relevant data is extracted from the web page.

Unicode Normalization

After the corpus is crawled it is subjected to unicode normalization. Unicode normalization ensures that the same character is not written using different unicode code points. For example in Tamil , கௌ in the word கௌரவம் can be typed as கௌ+ரவம் or கௌ+ௌ+ரவம் with two different encodings. Though the output looks similar, the second one is not the correct rendering. It needs to be normalized before processing. We have come up with an normalizer which normalises the spelling in Tamil.

Spelling Normalization

As we are crawling data from different sources, it is quite possible that the same word may be written using different spellings. We have not attempted to do spelling normalization in this current project. We would like to keep variations intentionally as the model should learn the lexical differences.

Similarly the wrong spelling in the text does not conform to canonical spelling or other grammatical rules of the language. In most situations it is desirable to preserve the error because taggers and parsers that learn their models from the data should learn how to deal with noisy input too. On the other hand, it is also desirable to mark such places as errors and to show the correct spelling, so that an application can hide bad sentences or present their correct version when necessary.²

Structural Richness in the Corpus

We have studied the different structures represented in the corpus. The corpus has a good coverage of simple sentences with nominal predicates, Intransitive, transitive and ditransitive verbs, passive construction, causative construction, non-nominative subject construction, Interrogative sentences. Similarly Compound sentences are found. Complex constructions which include complement clauses, participial clauses, conditional clauses and concessive clauses are also found. We have identified them using our initial version of morphological analyser looking at the suffix cue

4.2.2 Tokenization

Two types of tokenization is done on the data :

- (i) sentence tokenization and
- (ii) word tokenization.

Sentence tokenization

In Sentence tokenization the cleaned corpus is splitted into sentences based on sentence boundary markers like full stops, paragraph endings, question marks. In this stage for a given document each line will contain exactly one sentence.

Example:

Raw Corpus:

² <https://universaldependencies.org/u/overview/typos.html>

<https://www.tamilnadutourism.tn.gov.in/tamil/destinations/mahabalipuram-beach>
15-03-2023 14:24:28

பழமையான மற்றும் வரலாற்று தென்னிந்தியாவின் போக்கை வரையறுத்த பல வரலாற்று தருணங்களைக் கண்ட கடற்கரை இது. கடலோரம் காலப்போக்கில் பரிணாம வளர்ச்சியடைந்து இன்று தமிழ்நாட்டின் சுற்றுலா தலங்களில் ஒன்றாகும். மஹாபலிபுரம் ஒவ்வொரு ஆரவாரத்துக்கும் உரிய இடம். மகாபலிபுரம் வரலாற்றுச் சிறப்புமிக்க இடமாகும். பழங்கால கோயில்கள் முதல் அற்புதமான நினைவுச்சின்னங்கள் வரை, மகாபலிபுரம் சுற்றுலாப் பயணிகளுக்கு வாழ்நாளில் ஒருமுறை மட்டுமே அனுபவங்களை வழங்குகிறது. இந்த நகரம் அதன் புகழ்பெற்ற கடந்த காலத்திலிருந்து வெகுதூரம் வந்திருந்தாலும், அந்த காலத்தின் பிரதிபலிப்புகள் இன்னும் இப்பகுதியில் நீடித்து, அதை நாடி வருபவர்களை கவர்ந்திழுக்கிறது. இந்த நாட்களில் பரபரப்பான சுற்றுலா மையமாக உள்ளது.

Sentence Tokenized

www.tamilnadutourism.tn.gov.in/tamil/destinations/mahabalipuram-beach

15-03-2023 14:24:28

பழமையான மற்றும் வரலாற்று தென்னிந்தியாவின் போக்கை வரையறுத்த பல வரலாற்று தருணங்களைக் கண்ட கடற்கரை இது.

கடலோரம் காலப்போக்கில் பரிணாம வளர்ச்சியடைந்து இன்று தமிழ்நாட்டின் சுற்றுலா தலங்களில் ஒன்றாகும்.

மஹாபலிபுரம் ஒவ்வொரு ஆரவாரத்துக்கும் உரிய இடம்.

மகாபலிபுரம் வரலாற்றுச் சிறப்புமிக்க இடமாகும்.

பழங்கால கோயில்கள் முதல் அற்புதமான நினைவுச்சின்னங்கள் வரை, மகாபலிபுரம் சுற்றுலாப் பயணிகளுக்கு வாழ்நாளில் ஒருமுறை மட்டுமே அனுபவங்களை வழங்குகிறது.

இந்த நகரம் அதன் புகழ்பெற்ற கடந்த காலத்திலிருந்து வெகுதூரம் வந்திருந்தாலும், அந்த காலத்தின் பிரதிபலிப்புகள் இன்னும் இப்பகுதியில் நீடித்து, அதை நாடி வருபவர்களை கவர்ந்திழுக்கிறது.

இந்த நாட்களில் பரபரப்பான சுற்றுலா மையமாக உள்ளது.

Word tokenization

Word tokenization is carried out on each sentence in the document. Here each sentence would be splitted such that all punctuations are separated from the word, i.e a space is inserted between each punctuation mark. Now each sentence is splitted such that all tokens/words are separated. Here each file contains URL from where it is crawled, transliteration, sentence id followed by token number and the token itself.

For example:

```
# Sent_id = 1
# text = அழகும் கம்பீரமும் சங்கமிக்கும் இடம் இங்கே இயற்கை
அதிசயங்களும் மனிதனால் உருவாக்கப்பட்ட தலைசிறந்த படைப்புகளும்
மனதைக் கவரும் காட்சியை வழங்குகின்றன.
# transliteration = aḻakum kaṁpīramum caṅkamikkum iṭam iṅkē iyaṛkay aticayaṅkaḷum
maṇitaṅāḷ uruvākkappaṭṭa talayciṛanta paṭayppukaḷum maṇatayk kavarum kāṭciyay
vaḻaṅkukiṇṇa.
# url = https://www.tamilnadutourism.tn.gov.in/tamil/destinations/kanyakumari-beach
15-03-2023 14:24:09
1 அழகும்
2 கம்பீரமும்
3 சங்கமிக்கும்
4 இடம்
5 இங்கே
6 இயற்கை
7 அதிசயங்களும்
8 மனிதனால்
9 உருவாக்கப்பட்ட
10 தலைசிறந்த
11 படைப்புகளும்
12 மனதைக்
13 கவரும்
14 காட்சியை
15 வழங்குகின்றன
16 .

# Sent_id = 2
# text = இது ஒரே நேரத்தில் அழகாகவும், அழைக்கும் மற்றும்
கம்பீரமாகவும் இருக்கிறது.
# transliteration = itu orē nēratil aḻakākavum, aḻaykkum maṇṇum kaṁpīramākavum
```

```
irukkiratu.  
# url = https://www.tamilnadutourism.tn.gov.in/tamil/destinations/kanyakumari-beach  
15-03-2023 14:24:09  
1 இது  
2 ஒரே  
3 நேரத்தில்  
4 அழகாகவும்  
5 ,  
6 அழைக்கும்  
7 மற்றும்  
8 கம்பீரமாகவும்  
9 இருக்கிறது  
10 .
```

4.3 Annotation Process:

4.3.1 CONLL-U Format

The CoNLL-U format, short for "CoNLL Universal Dependencies," is a standard format for representing syntactic and morphological annotations in treebanks.

Here's a brief overview of the CONLLU format:

Columns:

Each field in a line represents a specific piece of information about the token. The standard CONLLU format consists of 10 columns:

ID (Column 1): A unique identifier for each token within the sentence.

FORM (Column 2): The actual word form of the token.

LEMMA (Column 3): The base or dictionary form of the word.

UPOS (Column 4): Universal part-of-speech tag.

XPOS (Column 5): Language-specific part-of-speech tag.

FEATS (Column 6): Morphological features.

HEAD (Column 7): The ID of the token's syntactic parent (head). The root token typically has a head value of 0.

DEPREL (Column 8): Syntactic dependency relation to the head.

DEPS (Column 9): Enhanced dependency graph in the form of head-deprel pairs.

MISC (Column 10): Additional miscellaneous information.

Example:

Here's an example of a sentence in CONLLU format in Tamil.

#Sent_id = 2

#text= நாம் சாப்பிடும் பழ வகைகளில் அனைத்து தேவையான சத்துக்களும் அடங்கியுள்ளன.

ID	Form	POS	Lemma	Morph	Relation	Head
1	நாம்	PRON	நாம்	Case=Nom Number=Sing Person=1	nsubj	2
2	சாப்பிடும்	VERB	சாப்பிடு	Polarity=Pos Tense=Fut VerbForm=Part	acl	4
3	பழ	NOUN	பழம்	Number=Sing	compound	4
4	வகைகளில்	NOUN	வகை	Case=Loc Number=Plur	obl	9
5	அனைத்து	DET	அனைத்து	—	det	7
6	தேவையான	ADJ	தேவையான	—	amod	7
7-8	சத்துக்களும்	-	-	-	-	-
7	சத்துக்கள்	NOUN	சத்து	Case=Nom Number=Plur	nsubj	9
8	உம்	PART	உம்	—	mark	7
9-10	அடங்கியுள்ளன	-	-	-	-	-
9	அடங்கி	VERB	அடங்கு	Polarity=Pos VerbForm=Conv	root	0
10	உள்ளன	AUX	உள்	Gender=Neut Number=Plur Person=3 Tense=Pres VerbForm=Fin	aux	9
11	.	PUNCT	.	—	punct	9

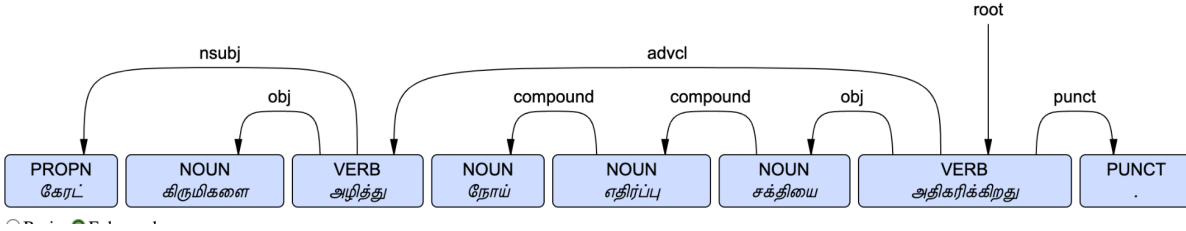
#Sent_id = 2

#text = கேரட் கிருமிகளை அழித்து நோய் எதிர்ப்பு சக்தியை அதிகரிக்கிறது.

#url = <https://www.femina.in/tamil/health/diet/a-carrot-a-day-1663.html> 03-04-2023 12:03:56

1	கேரட்	கேரட்	PROPN	—	Case=Nom Number=Sing	3	nsubj	3:nsubj	-
2	கிருமிகளை	கிருமி	NOUN	—	Case=Acc Number=Plur	3	obj	3:obj	-
3	அழித்து	அழி	VERB	—	Polarity=Pos VerbForm=Conv	7	advcl	7:advcl	-
4	நோய்	நோய்	NOUN	—	Number=Sing	5	compound	5:compound	-
5	எதிர்ப்பு	எதிர்ப்பு	NOUN	—	Number=Sing	6	compound	6:compound	-
6	சக்தியை	சக்தி	NOUN	—	Case=Acc Number=Sing	7	obj	7:obj	-
7	அதிகரிக்கிறது	அதிகரி	VERB	—	Gender=Neut Number=Sing Person=3 Tense=Pres VerbForm=Fin	0			
root	0:root	-							
8	.	.	PUNCT	—		7	punct	7:punct	-

The graph view is



4.3.2 Syntactic Tagset

Tamil Universal Dependency Relations

	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
conj cc	fixed flat compound	list parataxis	orphan goeswith	punct root dep

4.4. Annotation Tool:

For Annotation of data we have developed a User Interface such that an annotator can upload a file with data in CONLLU format. The tool will parse the data into sentences and furthermore each sentence into 10 columns viz token number, token, lemma, upos, xpos, morph analysis, head, relation, dependency relation, miscellaneous.

The tool facilitates the annotator to traverse the file sentence wise displaying only one sentence at a time. Users can view the dependency graph of the current sentence being displayed therefore enabling the user to make changes to the above mentioned columns like POS and dependency relation.

Major functionalities of the tool include editing any field in the CONLL data, multi word token splitting and merging. For the functionalities like token splitting and merging token numbers and relation numbers(head) field are automatically updated so that limiting the annotator's task to concentrate on actual annotation but not on mechanical tasks.

Below is the screenshot of the user interface being used to annotate the data.

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	MISC	Action
1	மெட்டபாவி	மெட்டபாவி	NOUN	_	Case=Nom N	7	nsubj	7:nsubj	-	INSERT DELETE
2	என்றால்	என்றால்	SCONJ	_	-	1	mark	1:mark	-	INSERT DELETE
3	மெட்டபாவி	மெட்டபாவி	NOUN	_	Case=Nom N	7	nsubj	7:nsubj	-	INSERT DELETE
4	என்பது	என்பது	SCONJ	_	-	3	mark	3:mark	-	INSERT DELETE
5	கமேளிகளை	கமேளி	NOUN	_	Case=Acc N	7	obj	7:obj	-	INSERT DELETE
6	உடல்	உடல்	NOUN	_	Case=Nom N	7	nsubj	7:nsubj	-	INSERT DELETE
7	எரிக்கின்ற	எரி	VERB	_	Polarity=Pos	8	acl	8:acl	-	INSERT DELETE
8	அளவு	அளவு	NOUN	_	Case=Nom N	0	root	0:root	-	INSERT DELETE
9	.	.	PUNCT	_	-	8	punct	8:punct	-	INSERT DELETE

SAVE

PREV 2 / 96 NEXT

RENDER TREE

In the above image we can see functions like upload file, navigate between sentences, view the graph, insert/delete some rows etc. After the annotation is completed users can download the file.

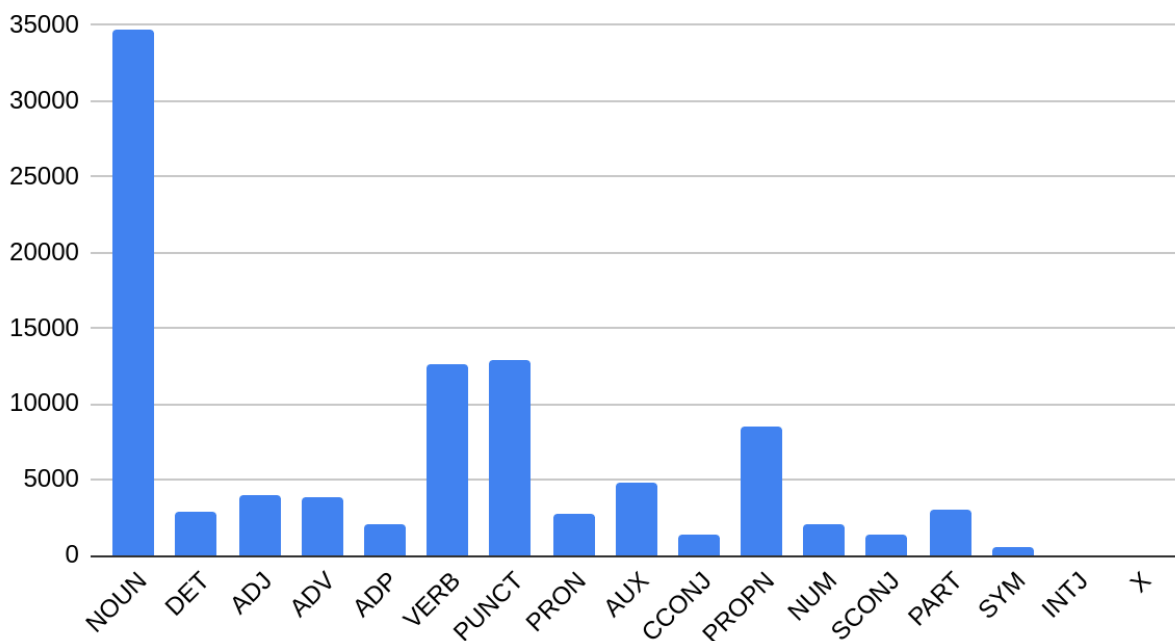
4.5. Statistics:

Parts of Speech Statistics in the data:

POS Category	Number of occurrences
NOUN	34758
DET	2951
ADJ	4022

ADV	3831
ADP	2111
VERB	12672
PUNCT	12907
PRON	2804
AUX	4820
CCONJ	1382
PROPN	8508
NUM	2084
SCONJ	1407
PART	3090
SYM	488
INTJ	3
X	58

Frequency analysis of POS tags in the data

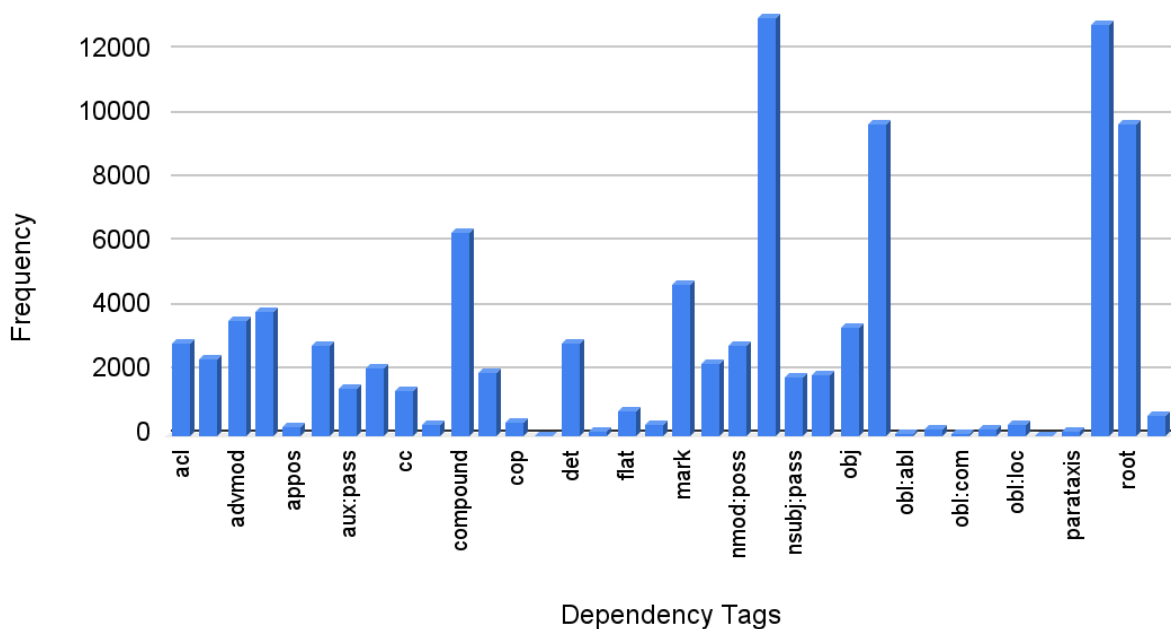


Syntactic relations frequency in the Data:

Relation	Number of occurrences
acl	2876
advcl	2389
advmod	3615
amod	3874
appos	304
aux	2787
aux:pass	1460
case	2097
cc	1382
ccomp	344
compound	6310
conj	1992
cop	447
csubj	19
det	2908
discourse	130
flat	766
iobj	309
mark	4708
nmod	2232
nmod:poss	2842
nsubj	13079
nsubj:pass	1852
nummod	1876
obj	3406
obl	9725
obl:abl	43

obl:agent	214
obl:com	40
obl:inst	222
obl:loc	313
obl:number	11
parataxis	131
punct	12851
root	9746
xcomp	602

Frequency analysis of Dependency Tags



Multi Word Tokens:

A total of 6K+ words are identified as multi word tokens in the data. These tokens are sometimes split into either 2 or 3 tokens following universal dependency guidelines.

Morph Analyzer:

A morph analyzer model is developed, extensive work is carried out to add paradigms, dictionary entries, derivations, and clitics. These morph features are converted to suit the universal dependency guidelines.

5. MODEL AND EVALUATION

5.1. Machine Learning Model

There are lots of off-the-shelf machine learning algorithms available in training parsers. One of the recent algorithm which reported with reasonable accuracy is Trankit (Van Nguyen et al, 2021). We customized it to the need of Tamil parsing. Trankit is a lightweight Transformer-based Toolkit for multilingual Natural Language Processing. It delivers a trainable pipeline for fundamental NLP tasks for over 100 languages and 90 pretrained pipelines for 56 languages. The Trankit toolkit outperforms prior multilingual NLP pipelines over sentence segmentation, part-of-speech tagging, morphological features tagging, and dependency parsing while holding competitive performance for tokenization, multi-word token expansion, and lemmatization. The memory usage and speed are efficient despite using large pretrained transformer models.

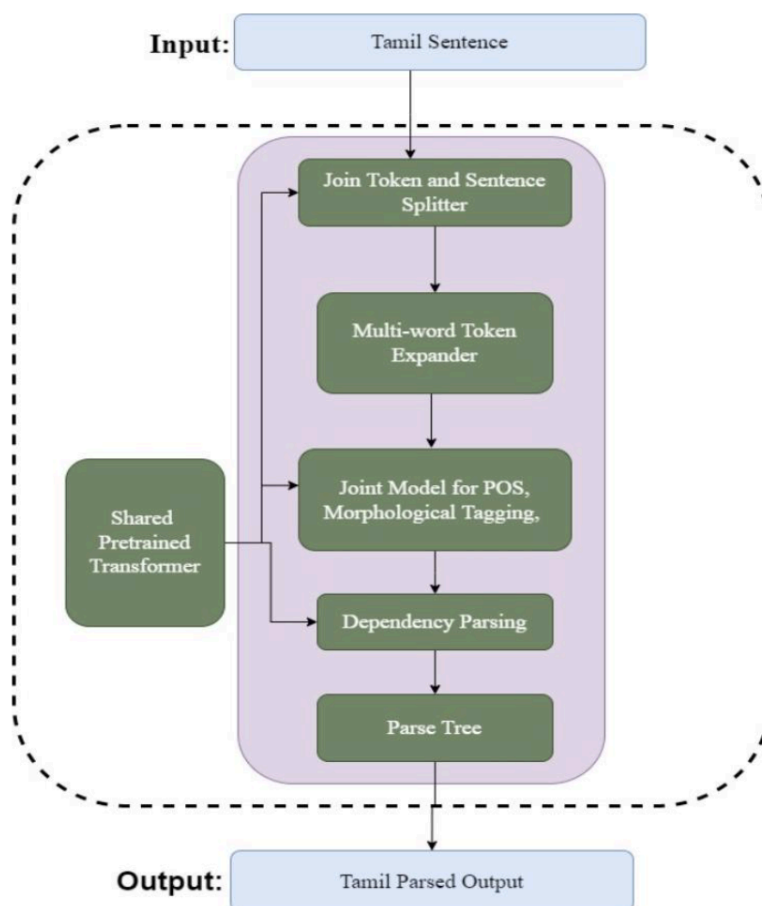


Figure: Architecture of Parsing

As shown in the figure, the overall architecture of the Trankit pipeline consists of three novel transformer-based components, Joint token and sentence splitter and the joint model for POS tagging, morphological tagging, dependency parsing.

The main concern of the toolkit involves GPU memory, where different transformer-based components in the pipeline for one or multiple languages must be loaded simultaneously to serve multilingual tasks. They have introduced a plug-and-play mechanism with Adapters to address this memory issue. Adapters are small networks injected inside all layers of the pretrained transformer model that have shown their effectiveness as a lightweight alternative for traditional finetuning of pretrained transformers. The adapter and task-specific weights are only updated during training. At inference time, depending upon the language of the input, the corresponding trained adapter and task-specific weights are activated and plugged into the pipeline for processing

5.2 Evaluation

Metrics for Tokens, Lemma, UPOS, UFeats is calculated using F1-score and for dependency tags UAS i.e Unlabelled attached score and Label attachment score is calculated.

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

6. TAMIL PARSER GUI AND BACKEND

6.1 Introduction:

For this task we have chosen Next.js for user interface as it is a simple yet powerful React framework. It makes building websites easier with features like server-side rendering and automatic code splitting.

6.2 Overview:

The Application is divided into server side and user interface. On the server side, the API calls are managed to get the Parser output from the machine learning models which will be passed onto the client side code to render on the browser.

6.3 Backend Model:

The backend model is powered by Trankit, a machine learning framework. We have fine tuned our data on the Trankit model and the model is integrated to the server as a API call such that when an API call is made to the server, the input sentence is passed on to the model and the parser output that includes all 10 fields of CONLLu data is given by the model.

6.4 Client Side user interface

As explained earlier we have used Next.js for this task. When the user inputs a sentence in the input area, it is passed onto the API and the response from the API call is shown in different forms such as POS, Syntax View, Tree View, Morph analysis view. A dependency graph is also shown for better understanding. In subsequent sections we explain different options provided in the user interface.

6.4.1 Parser:

Graph: Visualize your data through an interactive graph, comprehensively representing relationships and patterns within the provided information.

POS Tags (Part-of-Speech Tags): Gain linguistic insights into your text by extracting POS tags. This feature categorizes words in your input, enhancing your understanding of the language structure.

Morphological Analysis: Uncover the intricacies of language by receiving a morphological analysis of your text. This feature breaks down words into their root form and other grammatical features providing a deeper understanding of linguistic elements.

6.4.2 Graph view:



Where data visualization meets linguistic analysis! In this section, you can input text in CoNLL-U format, and our application will dynamically generate a graph to represent the underlying structure.

6.4.3 Guidelines:


Three key guidelines: POS (Part-of-Speech), Morph Analysis, and Syntactic Analysis. Each menu item in the guideline section is clickable that will open corresponding guidelines which are selected by the user.

6.5 User Interface screenshots:

6.5.1 Tamil Syntactic Parser screenshot



தமிழ் தொடரியல் பகுப்பாய்வி
Tamil Syntactic Parser



- GraphView
- Guidelines
- Parts of Speech
- Morph Analysis
- Syntactic Analysis

தொடரியல் பகுப்பாய்வி சொற்றொடர்களின் உள்ளமைப்பை ஆராயும் ஒரு தானியங்கி. இது சிக்கலான தொடர் அமைப்புகளை சிறிய சொற்கூறுகளாகப் பிரித்தாய்ந்து, அதன் உள்ளமைப்பு மற்றும் சொற்களுக்கு இடையேயான தொடர்பை வெளிப்படுத்துகிறது.

A parser is an automated tool that provides the internal structure of sentences. It decomposes complex structures into their constituent parts and represents relations between words.

சென்னை தமிழ்நாட்டின் தலைநகரம் ஆகும்.

Options

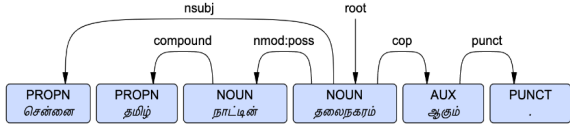
தொடர் பகுப்பாய்வு (Syntactic Analysis) சொற் கூறுகள் (Parts of Speech) உருபன் பகுப்பாய்வு (Morph Analysis)

பகுப்பாய்வு (PARSE)

Sentences

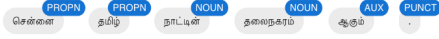
சென்னை தமிழ்நாட்டின் தலைநகரம் ஆகும்.

Syntactic Analysis



```
graph TD
    root((root)) --- nsubj[nsbj]
    root --- nmod[nmod:poss]
    root --- cop[cop]
    root --- punct[punct]
    nsubj --- PROPN1[PROPN  
சென்னை]
    nsubj --- PROPN2[PROPN  
தமிழ்]
    nmod --- NOUN1[NOUN  
நாட்டின்]
    nmod --- NOUN2[NOUN  
தலைநகரம்]
    cop --- AUX[AUX  
ஆகும்]
    punct --- PUNCT[PUNCT  
.]
```

Parts of Speech



சென்னை (PROPN) தமிழ் (PROPN) நாட்டின் (NOUN) தலைநகரம் (NOUN) ஆகும் (AUX) . (PUNCT)

Morph Analysis

Word	POS	Lemma	Morph
சென்னை	PROPN	சென்னை	Case=NomiNumber=Sing
தமிழ்	PROPN	தமிழ்	Case=NomiNumber=Sing
நாட்டின்	NOUN	நாடு	Case=GeniNumber=Sing
தலைநகரம்	NOUN	தலைநகரம்	Case=NomiNumber=Sing
ஆகும்	AUX	ஆடு	Gender=NeutlNumber=SinglPerson=3lTense=FutlVerbForm=Fin
.	PUNCT	.	-

Funded By [Tamil Virtual Academy\(TVA\)](#) Principle Investigator [Parameswari Krishnamurthy](#) Under the Project **A Syntactic Parser for Tamil(2023 - 2024)**

6.5.2 Guidelines screenshot

தமிழ் தொடரியல் பகுப்பாய்வி

Tamil Syntactic Parser

- Parser
- GraphView
- Guidelines
- Parts of Speech
- Morph Analysis
- Syntactic Analysis

TAMIL PARTS-OF-SPEECH TAGGING GUIDELINES

Designing Targets and Specifications

Part of speech (POS) tagging is the process of assigning a unique part of speech to each word (token) in a sentence. This process helps in identifying and disambiguating the role of each word (token) in a sentence. POS tagger tags words in a context using POS tags.

This POS guideline 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 morphological, syntactic and semantic structure of the sentence. It has described a set of 15 tags with examples as seen below:

Open class words	Classed class words	Other
ADJ	ADP	PUNCT
ADV	ATX	SYM
INFL	CONJ	
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	

1. ADJ - adjective
Adjectives are words that modify nouns. The following are various occurrences of ADJ in Tamil. The word ends with ஆள் are tagged adjectives, describing various features as follows.

01. Attributive adjective: Attributive adjectives provide additional information about a noun by describing its characteristics. It occurs before the noun.

சுவை பழமாலைகளையுடைய
வெள்ளை வெள்ளை தோல்கொண்டவருடைய
பெண்கள் அவர்கள்.

Funded By [Tamil Virtual Academy\(TVA\)](#)
Principle Investigator [Parameswari Krishnamurthy](#)
Under the Project [A Syntactic Parser for Tamil\(2023 - 2024\)](#)

6.5.3 Graph view screenshot

தமிழ் தொடரியல் பகுப்பாய்வி

Tamil Syntactic Parser

- Parser
- GraphView
- Guidelines
- Parts of Speech
- Morph Analysis
- Syntactic Analysis

sent_id = test-s1

text = பிகாரிலிருந்து ஏராளமான இளைஞர்கள் வேலை தேடி வெளி மாநிலங்களுக்கு குடிபெயர்ந்து வருகின்றனர்.

translit = pikāriliruntu ērālamāna [laiŋarka] vēlai tēi vēli mānilaŋkalukku kuṭipēyartu varukinranar.

orig_file_sentence 001#1

1-2 பிகாரிலிருந்து - - - - - Case=NomiGender=NeutNumber=SinglPerson=3 4 nmod 4:nmod:இலிருந்து:nom LTranslit=pikāriTranslit=pikār

2 இலிருந்து இலிருந்து ADP PP----- AdpType=Post 1 case 1:case LTranslit=iliruntuTranslit=iliruntu

3 ஏராளமான ஏராளமான ADJ JJ----- 4 amod 4:amod Translit=ērālamānaLTranslit=ērālamāna

4 இளைஞர்கள் இளைஞர் NOUN NNN-3PA-- Animacy=AnimlCase=NomiGender=ComlNumber=PluriPerson=3 9 nsubj 9:nsubj Translit=ilaiŋarkaLTranslit=ilaiŋar

5 வேலை வேலை NOUN NNN-3SN-- Case=NomiGender=NeutNumber=SinglPerson=3 6 obj 6:obj Translit=vēlailLTranslit=vēlai

UPLOAD FILE
PARSE

sent_id = test-s1

text = பிகாரிலிருந்து ஏராளமான இளைஞர்கள் வேலை தேடி வெளி மாநிலங்களுக்கு குடிபெயர்ந்து வருகின்றனர்.

translit = pikāriliruntu ērālamāna [laiŋarka] vēlai tēi vēli mānilaŋkalukku kuṭipēyartu varukinranar.

orig_file_sentence 001#1

Funded By [Tamil Virtual Academy\(TVA\)](#)
Principle Investigator [Parameswari Krishnamurthy](#)
Under the Project [A Syntactic Parser for Tamil\(2023 - 2024\)](#)

7. CONCLUSION

As a part of the project, milestone-4 is met with evaluation of Tamil Parser and final documentation. The annotation of 100K Tamil tokens for its POS, morph features, and syntactic tags are completed. Guidelines for POS tagging, Morphological feature identification, syntactic tags for building treebank, and multi-token word identification are prepared and revised. These guidelines are being used while tagging Tamil sentences to build a syntactic parser. Evaluation is done at each stage of model building to analyze the results produced.

REFERENCES

- De Marneffe, M.C., Manning, C.D., Nivre, J. and Zeman, D., 2021. Universal dependencies. *Computational linguistics*, 47(2), pp.255-308.
- Hess, C.W., Ritchie, K.P. and Landry, R.G., 1984. The type-token ratio and vocabulary performance. *Psychological Reports*, 55(1), pp.51-57.
- Parameswari Krishnamurthy and Sarveswaran K. 2022. **Towards Building a Modern Written Tamil Treebank**. In Proceedings of the *20th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2021)*, Sofia, March 21-25, 2022. ACL Anthology.
- Parameswari Krishnamurthy and Sarveswaran K. 2021. *Multi-Word Token Expansion for Tamil: A Parsing Perspective*. Paper presented at Tamil Internet conference-2021, 2-5 December 2021, Virtual.
- Parameswari Krishnamurthy, Keerthana B., Sangeetha P. 2021. *A Rule-Based Dependency parsing for Tamil*. In ``Tamil Computing - Tools and Applications Young Researchers' Conference 2021 (TaCTA-YRC2021)" 12-13 March.
- Sangeetha, P., Parameswari Krishnamurthy and Amba Kulkarni. 2021. *A Rule-based Dependency Parser for Telugu: An Experiment with Simple Sentences*. Translation Today. Volume 15 Issue 1, July 2021.
- Sangeetha P. Parameswari Krishnamurthy and Amba Kulkarni. 2021. Parsing Subordinate Clauses in Telugu using Rule-based Dependency Parser. In proceedings of *Workshop on Parsing and its Applications for Indian Languages (PAIL)*, co-located with ICON 2021, 16 December 16, 2021. ACL Anthology.
- Sato, Motoki, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. "Adversarial training for cross-domain universal dependency parsing." In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 71-79

/END.

FUND BUDGET BREAK-UP AFTER GST:

Head	Sanctioned Amount (Before GST clarification)	Sanctioned Amount (After GST clarification)	1st Receipt
Manpower	8,44,000	7,15,254	2,14,576
Contingency	20,000	16,949	5,085
Travel and Publication	50,000	42,373	12,712
Consultancy	50,000	42,373	12,712
	9,64,000	8,16,949	2,45,085
Overheads (9.9585 %)	96,000	81,356	24,407
Total	10,60,000	8,98,305	2,69,492
GST @ 18% on project cost		1,61,695	48,508
Total amount		10,60,000	3,18,000

1st Installment (after 1 Milestone)	3,18,000 (Received)
2nd Installment (after 2 Milestone)	3,18,000 (Received)
3rd Installment (after 3 & 4 Milestone)	2,12,000
4th Installment (after confirming Usage)	2,12,000

	10,60,000
--	------------------