CODIFICATION OF NAVALAR'S TAMIL GRAMMAR

K. Sarveswaran[1*] and S. Mahesan[2]
[1, 2]Department of Computer Science, University of Jaffna

Abstract

We have described the codification process of Navalar's Tamil grammar. In this process, JSON (JavaScript Object Notation) is used to store linguistic information, and Python is used to write linguistic rules. Since Tamil is a low-resource language, it is essential to develop tools and resources and make them available for technologists and linguists to take up Tamil language computation to the next level. This paper gives an account of Tamil grammar tradition, a note on Navalar's Tamil grammar, the approach used to codify the grammar, and the challenges faced when codifying Navalar's grammar.

Keywords: *Codifying grammar, Grammar modelling, Low-resource language, Navalar's grammar, Tamil grammar*

**Corresponding author: Email: sarves@univ.jfn.ac.lk; Tel: 772244192*
https://orcid.org/0000-0003-1579-0597

Introduction

The development of Tamil grammar has a long history. *tolkāppiyam* is identified as the very first scholarly work on Tamil grammar. The date of this publication is not exact, yet it is believed to have been published more than 2500 years ago. *tolkāppiyam* is considered to be a derived work of an even older work called *agastyam*; the whole work of *agastyam* is not extant. Tamil grammar publications can be divided into ones composed by native scholars and those that Europeans have written to facilitate the acquisition of the Tamil language, mainly by foreigners. Native grammarians prescribe rules for phonology, orthography, morphology, syntax, and semantics of the Tamil language based on the literature found previous to their period. Grammars prescribed by native scholars are considered to be the standard for writing in Tamil and are taught in schools. Although there are a few works like *aṭippaṭait tamiḻ ilakkaṇam*, most of the other grammars taught in schools are derived from ancient texts. The grammar that is still widely used and taught is what is defined in *nannūl*, which dates back to the 13th century and is derived from *tolkāppiyam.* Therefore, it is challenging to use this grammar to carry out linguistic analysis of modern Tamil texts, which have evolved significantly over time, especially with the boom of the Internet.

This is the context in which we have attempted to codify a recent Tamil grammar from the book *ilakkaṇac curukkam* written by *śrīlaśrī āṟumukanāvalar,* which is a derived work of *nannūl*. We have outlined the approach and the challenges we faced when codifying the Navalar's grammar. It is important to note that the year 2022 is declared as the 'Navalar year' by the government of Sri Lanka.

Need for codifying a Tamil grammar

Tamil is a low-resource language in terms of computational resources such as annotated data and language processing tools. On the other hand, the state-of-the-art approach to language computation requires a significant amount of annotated data for training, development, and testing. Although several natural language processing toolkits, such as NLTK[1] are available publicly, none of them have support for Tamil linguistic processing. Since Tamil is distinct from European languages and other resourceful languages, the existing resources cannot be used straight away. In addition, analyzing a large amount of Tamil text would be challenging due to the rich nature of Tamil. On the other hand, more and more applications are being developed in local languages, and the local language adaptation has become an important unique

[1] https://www.nltk.org/

selling proposition in the business world. Therefore, it is essential to develop computational resources for the Tamil language to support linguistic analyses and language computation.

Materials and Methods

Navalar's Tamil grammar

Arumuka Navalar alias Nallur Arumuga Pillai (1822-1879) was a Tamil language scholar, polemicist, and religious reformer who has made ninetyseven publications, including the book *ilakkaṇac curukkam* for Tamil grammar. This work has 406 statements or rules divided into three sections, namely *eḻuttatikāram* (orthography, phonology, and morphology), *collatikāram* (morpho-syntax), and *toṭarmoḻiyatikāram* (syntax).

The first section - *eḻuttatikāram* introduces the Tamil alphabet and various rules that need to be followed when writing letters together. This section also briefly covers how words are formed by introducing different parts of a word. A large portion of this first section is *puṇariyal* which outlines the phonological rules that must be used when conjugating two morphs or compounding two words.

The second section - *collatikāram* covers the four types of words, namely nouns, verbs, particles and pre-/postpositions, and adjective/adverbs and their morpho-syntactic properties and semantics.

The third section - *toṭarmoḻiyatikāram* is the short section which outlines the types of phrases and simple sentences and their syntax, as well as some common errors occurring in texts.

The coverage of this grammar is not comprehensive enough to model the Tamil language using formal approaches. For instance, the interactions between morphology and syntax are important in identifying grammatical cases and subordinate clauses. However, such linguistic phenomena are not discussed in Navalar's grammar.

Our approach

There are several approaches used to capture the linguistic information of a language. For instance, grammar formalisms such as TAG, LFG, HPSG, and CCG or computational formalisms such as Finite-State Machines are used to capture linguistic information. Apart from these formal approaches, linguistic information is also captured using computer data structures and programs, like NLTK. The former requires a very in-depth and comprehensive descriptive study of a language using modern linguistic theories. A few initial attempts have been made to use formal approaches to model Tamil grammar.

Nalvalar's grammar is in the form of rules, and these rules are not comprehensive enough to model the Tamil language. Therefore, in this paper, we have attempted to codify part of Navalar's grammar using data structures and programs and published them as application programming interfaces (APIs) for others to make use of the knowledge. These APIs can be used to do language preprocessing and develop other language applications. For instance, this can be used to validate texts before carrying out any morphological analysis or preprocess and clean text before training a machine translation system. Apart from its usage in the natural language processing domain, this API is also useful for carrying out linguistic analyses.

We have used JSON[4] to store information and Python[5] to write rules. For instance, the list of letters with which a word can start is a fixed number of letters, and they are stored in a JSON object. We have used Python rules to check if the initial letter is valid, as this cannot be directly checked due to the complex nature of Tamil orthography.

Conclusions and Recommendations

In this 'Navalar year', we have attempted to codify Navalar's Tamil grammar in view to help technologists process the Tamil language and linguists carry out analyses. We use JSON to store linguistic data and Python rules to process and analyse information.

We encountered several challenges when codifying the grammar, and because of that, we could not cover the grammar entirely. For instance, entities in Tamil (nouns) are primarily categorised as rational and irrational, which is necessary to write rules. However, compiling the words in a language and categorising them is challenging and tedious. Further, Navalar's grammar is written based on texts from the 18th century; therefore, the grammar does not capture the modern grammatical constructs and lexical words found in recent texts. However, we are in the process of completing the codification using other means, like by using machine learning approaches.

Although the codified Navalar's grammar is not comprehensive, it is helpful to carry out basic linguistic analyses and language preprocessing tasks. More importantly, this attempt is an appropriate contribution to Nalvalar's year 2022.

[4] https://www.json.org/json-en.html

[2] https://www.python.org/

References

Navalar, A. (1921). *Ilakkaṇac curukkam* (in Tamil). Vittiyānupālaṉa Yantiracālai.

Sarveswaran, K., & Butt, M. (2019). Computational Challenges with Tamil Complex Predicates. In M. Butt, T. H. King, & I. Toivonen (Eds.), *Proceedings of the 24th International LFG Conference (LFG2019), Australian National University* (pp. 272–292). Retrieved from http://cslipublications.stanford.edu/LFG/2019/lfg2019-sarveswaran-butt.pdf

Sarveswaran, K., & Mahesan, S. (2014). Hierarchical Tag-set for Rule-based Processing of Tamil Language. *International Journal of Multidisciplinary Studies (IJMS),* *1*(2), 67-74. https://doi.org/http://doi.org/10.4038/ijms.v1i2.53

Sarveswaran, K., Dias, G. & Butt, M. (2021). *Thamizhi*Morph: A morphological parser for the Tamil language. *Machine Translation*, 35, 37–70. https://doi.org/10.1007/s10590-021-09261-5

Shanmugathas A. (1979). moḻiyiyal nōkkil nāvalar (in Tamil), In K. Kailasapathi (Ed.), *Arumuga Navalar Centuary Volume* (pp. 27-35). Srilasiri Arumuga Navalar Sabai.