Erschienen in: 2018 3rd International Conference on Information Technology Research (ICITR). - Piscataway, NJ : IEEE, 2018. - ISBN 978-1-72811-470-5
https://dx.doi.org/10.1109/ICITR.2018.8736139

ThamizhiFST: A Morphological Analyser and Generator for Tamil Verbs

K Sarveswaran
Dept. of CSE
University of Moratuwa
Moratuwa, Sri Lanka
sarvesk@uom.lk

Gihan Dias
Dept. of CSE
University of Moratuwa
Moratuwa, Sri Lanka
gihan@uom.lk

Miriam Butt
Dept. of Linguistics
University of Konstanz
Konstanz, Germany
miriam.butt@uni-konstanz.de

***Abstract*—ThamizhiFST is a Morphological Analyser and Generator (MAG) for Tamil. It was developed to extend the coverage of the computational Tamil grammar being developed using Lexical Functional Grammar (LFG). ThamizhiFST covers the simple verbs in Tamil as an initial step. A Finite State Transducer (FST) approach was used to develop the MAG and it was implemented using the FOMA Open Source Software. Since morphological rules are of a finite nature and represent a known quantity, a rule-based approach like FST is more appropriate than possible machine learning alternatives, especially with respect to achieving reliably good accuracy that is required for computational grammar development. A set of 3250 Tamil verb lemmas from 13 paradigms together with their 260 conjugation forms were used in the construction of ThamizhiFST. Further, a set of 27 labels were used to mark the morphosyntactic information of the verbs. The whole system was developed as a three-layer web-based system to tackle the issues arising when processing an agglutinative language like Tamil and to ensure its extendability. Unlike other existing MAGs, ThamizhiFST also provides the morpheme corresponding to each morphosyntactic label and marks morpheme boundaries. An evaluation shows that ThamizhiFST has an f-measure of 0.97 for simple verbs. Future and current work include work on extending the system to cover more verbs and nouns and make it generally available.**

***Index Terms*—morphological analyser, morphological generator, finite state transducer, Tamil**

I. Introduction

A morphological analyser can analyse a given word and find the lemma and its morpheme information. A morphological generator generates or synthesises the surface form from the given lemma and morpheme information. These two are crucial tools for Natural Language Processing (NLP), especially for processing morphologically rich languages where crucial syntactic and semantic information is expressed via morphological means. Morphological analysers and generators (MAG) are widely used in NLP applications such as machine translation, grammar engineering, spell checking, question-answering systems, information mining, and Computer Assisted Language Learning. This paper outlines the development of a MAG called ThamizhiFST for Tamil verbs in view of supporting grammar engineering while considering its extendability and potential for wide usage.

II. Background

A. The Tamil language

Tamil is a Dravidian language, specifically a South Dravidian language that is spoken by more than 80 million people across the world as their mother tongue. It has also been recognised as a classical language by the government of India as it has more than 2000 years of a continuous, unbroken, literary tradition [1]. Further, it is an official language of Sri Lanka and Singapore, and has regional official status in Tamil Nadu and Pondichchery, India. It has also been recognised as a minority language or indigenous language in several countries including Malaysia, Mauritius and South Africa. Finally, Tamil is taught as a second language in several other countries including Canada, Australia and the United Kingdom.

B. Morphology of Tamil

Tamil is an agglutinative language where a set of morphemes are generally suffixed to a lemma. However, there are a few instances in which morphemes are prefixed to lemmas in Tamil. For instance, in the word அநியாயம் - aniyāyam - 'injustice' the அ is used to mark negation. However, not all the words that start with the letter அ are negative words. Tamil words naturally can take inflectional and derivational suffixes. The nouns in Tamil are primarily marked for case and number. In contrast, Tamil verbs display complex morphological paradigms that express a range of information relevant for syntactic and semantic analysis.

C. Finite State Transducers

The theory of two level morphology saw successful early applications for morphologically rich languages like Finnish, Russian and Sanskrit [2]. Subsequently, it was taken up by researchers to develop morphological analysers for many other languages, including South Asian languages like Urdu [3], [4].

In two level morphology, a word is represented at two levels, namely the lexical level and the surface level. For instance, the English word *cats* would be taken as the surface form and be related to the lexical form `cat+N+PL`, which contains an analysis in terms of lemmas and morphological tags [5], [6]. Finite State Transducers (FST) have been successfully used to implement the ideas behind two level morphology so that when the system receives a surface form as input, the given finite automaton will generate the lexical form and vice versa (input in terms of a lexical form results in the output of a surface form); such an automaton is called a Finite State Transducer [6].

III. Tamil verbs

Verbs in Tamil comprise a range of morphological information including tense, mood, aspect, negation, interrogation, emphasis, speaker perspective, sentience/rationality, and conditional and causal relations [7].

Entities in Tamil are fundamentally classified into rational vs. irrational. Entities are termed rational if they are perceived as being able to think on their own, otherwise those entities are termed irrational.

Some researchers also claim that the weak vs. strong distinction found in the verbal paradigms can be used to determine transitivity, ergativity, volitativity and affectedness [8]. All of this information is indicated by morphemes that are suffixed to a verb lemma. For instance, *he had been coming* can be written in Tamil as வந்துகொண்டிருந்திருக்கிறான் 'vantukoṇṭiruntirukkiā'. This consists of the following morphemes: வா (lemma: 'come') + கொண்டு (continuous) + இருந்து (has) + இரு (be) + கிறு (present tense)+ ஆன் (3rd person + Singular + Masculine + Rational). In Tamil, person, number, gender and rationality are marked using only one morpheme if those are available in verbs. For instance, the morpheme ஆன் marks all of this information in the above example.

A. Classification of verbs

Tamil verbs can be classified based on criteria coming from morphology, syntax and semantics [8], [9]. Many Western scholars have classified verbs based on the morphology, specifically, based on how morphemes conjugate [10], [11], [12]. There is an initial classification of verbs by Graul [11] which other scholars have followed, which remains widely accepted, and which was adapted for the Tamil lexicon project [13]. In that approach the Tamil verb lemmas are classified into twelve categories based on the future tense markers of the verbs. Irākavaiyaṅkār [14] has compiled Tamil verbs from the literature up until the year 1958 and classified them based on Graul's [11] scheme [15].

IV. ThamizhiFST: A Finite State Morphological Analyser for Tamil

The primary approaches for building morphological analysers include rule-based, machine learning and hybrid combinations. Each of these major approaches have their own strengths and weaknesses. The implementation of an FST is the technique primarily used in the rule-based approach. It was decided to use FST to develop the ThamizhiFST as this methodology ensures high accuracy that is necessary for grammar engineering and robustness can be ensured via a guesser module [5]. FST based MAGs can also be easily adapted and ported to different domains. According to [6] lexicon, morphotactics and orthographic rules are the key elements of a FST based MAGs. These have been implemented as part of ThamizhiFST.

A. Need for the ThamizhiFST

Tamil is morphologically complex and the morphology provides information that is essential for further syntactic and semantic processing. As such, a MAG for Tamil is needed. Since Tamil verbs are particularly morphologically complex, the implementation began by focusing on Tamil verbs. There have been several previous attempts at the development of a morphological analyser and/or generator for Tamil, two of which were based on machine learning [16], [17] and FST [18]. Unfortunately none of these are currently active or available in the public domain. The extendability or re-usability of those tools are thus questionable because of the non-availability of the data and code base.

For our particular purpose, namely grammar engineering, a morphological analyser with good accuracy is necessary. Furthermore, the computational Lexical Functional Grammar (LFG) grammar that is being developed uses the grammar development platform XLE [1] and the insights of the ParGram effort [19]. Grammars developed with XLE are most easily interfaced with an FST based morphological analyser. Finally, as we aim to use the default tokeniser provided by XLE in the grammar, we need to ensure that the output of the MAG is in a specific form expected by the XLE. Without access to the code and data of the existing MAGs, it is therefore difficult to develop a MAG from existing resources to support our grammar engineering effort.

B. Lexicon

A lexicon of 3,250 lemmata was compiled from the following two verified sources:

- Cre-A publishers [7] have identified 369 verbs that are used in modern Tamil. The authors have followed a well structured approach for the identification of these 369 verbs. They have analysed a corpus of 7 million tokens. Further, they have obtained expert advice to compile the list. This list now has been included in the contemporary Tamil dictionary called Cre-A [20].
- Irākavaiyaṅkār [14] has surveyed Tamil literature up until 1958, has identified 3124 lemmas and has categorised those into 12 classes as per Graul's classification [15]. Although some of these verbs are no longer

[1] http://www2.parc.com/isl/groups/nltt/xle/

current, they have been included in the development of the ThamizhiFST in order to be allow for the processing of historical Tamil text.

There are tagged Tamil corpora available on the internet. However, since the tagging is not done manually and / or not thoroughly verified, one cannot rely on the accuracy of these resources. Therefore, it was decided to use only the verbs compiled from the above mentioned two verified sources. Later, more lemmata will be included from tagged corpora after a validation process.

Cre-A [7] has also identified 254 forms for each Tamil verb after a rigorous analysis of contemporary texts. Some verbs may not take all of the 254 forms. Further, Rajaram [13] has also identified 21 forms for each verb from a pedagogical perspective. On the other hand, it has been claimed that a Tamil verb lemma can take up to 8000 forms [17].

In this study only 260 forms were considered. These forms are the set that is the common set of [7] and [13]. For each lemma these 260 forms can be generated and analysed. Therefore, the ThamizhiFST can analyse or generate around 800,000 verb surface forms. However, some of the generated forms by the ThamizhiFST may not exist, at least in the contemporary usage. However, this factor is not considered at this point of development.

1) Verb Paradigm: A research group [16] has proposed a 32 paradigm approach in their data driven morphological analyser for Tamil verbs while another study [21] proposes a 34 verb paradigm for Tamil verbs. However, the widely accepted Graul's [11] 12 verb paradigm is used in this research. In addition to these 12 categories, one more category for irregular verbs was added. All the verbs that do not fit into those 12 categories were handled in this 13^{th} category. The irregular verbs listed by the [7] are also included in this category.

C. Morpheme labels

There are different sets of labels used in the morphological analysers to mark the morphemes. There is an attempt [22] to unify the morphological labels in the Unimorph project so that crosslingual morphological transfer becomes easier. However, in the ThamizhiFST, we have developed a set of our own morpheme labels that are listed in the Table I.

Because person, number, gender and rationality are marked by a single morpheme in Tamil, it is more efficient from a grammar engineering perspective to handle them together and it will also reduce the number of lexical rules in the grammar [23]. Therefore, it was decided to develop our own labeling rather than adopting that of the Unimorph. However, we will implement a feature as part of ThamizhiFST that will facilitate exporting information in the Unimorph format as well.

A paper [8] argues that weak or strong cannot be used to determine transitivity, volitivity, affectedness and ergativity with many case examples though some other authors believe so. Therefore, in this study only the morphologically weak or strong nature of the verb is marked.

TABLE I
Morpheme labels & Description

Morpheme label	Description
verb	Part of Speech
past	Past tense
pres	Present tense
fut	Future tense
1sg	1st person - singular
1pl	1st person - plural
2sg	2nd person - singular
2sgh	2nd person - singular - honorific
2pl	2nd person - plural
weak	Weak verb
strong	Strong verb
euph	Euphonic marker
neg	Negation
imp	Imperative
inf	Infinitive
relPart	Relative Particle
verbPart	Verbal Particle
cond	Conditional
causal	Causal
temp	Temporal
mood	Mood
aspect	Aspect
caus	Causative
temp	temporal
opt	Optive
con	Concessive
vNoun	Verbal Noun

D. Challenges in writing orthographic rules

In most cases the suffixes are not just added to a lemma. Instead, several orthographical changes happen during the suffixation in Tamil due to grammatical and phonological reasons. These make the orthographic rules complex in Tamil. The following are the common orthographical changes observed during the suffixation and that are handled in ThamizhiFST:

- Morpheme/s can be just suffixed to the lemma, however, when there is a consonant followed by a vowel those two together become composite. For instance, சய் - 'Cey' - do + த் (past tense marker) + ஆன் (third person, masculine, singular and rational) = சய்தான் - 'Ceytā' - 'he did', where த் + ஆ together form the composite character தா.
- A new letter is introduced in addition to the morpheme. For example நட - 'Naṭa' - 'walk' + த் - (past tense marker) + ஆன் (third person, masculine, singular and rational) = நடந்தான் - 'Naṭantā' - 'he walked', where a letter ந் is introduced. Some researchers consider this as a Sandhi letter, but some modern linguists consider ந்த் as the past tense marker [24].
- Two letters together can form a new letter during the suffixation. For example கொள் - 'Koḷ' - 'take' + த்

(past tense marker)+ ஆன் (third person, masculine, singular and rational) = கொண்டான் - 'Koṇṭā' - 'he took', where ள்+த் becomes ண் .

- A new morpheme called an euphonic morpheme or a sound filling particle, can be introduced in addition to required morphemes. For instance in the word செய்தனம், செய் - 'Cey' - 'do' + த் (past tense marker) + (அன் (euphonic marker)) + அம் (first person, neuter, plural and rational), the அன் which is in the parenthesis is an example for a sound filling particle or euphonic marker.
- Irregular verbs in Tamil may undergo a complete change in its surface form when conjugate. தா 'give' take different forms such as தா / தரு / த. For instance, in past tense forms the lemma becomes த, in present and future tense forms it becomes தரு and the imperative form would be தா. For example, தந்தான் - 'Tantā' - 'he gave', தருவான் - 'Taruvā' - 'he will give', and தா - 'Tā' - 'give'.
- A consonantal glide may be introduced when there are two consecutive vowels. Tamil has two such consonantal glides, e.g., ய் and வ்.

E. Design and architecture

Wide applicability and ease of extendability were also key considerations in the development of ThamizhiFST. Therefore, not only the morpheme labels but the actual morphemes are also stored. Further, it was decided to process the text in form of Unicode instead of as Romanised text as done in other morphological analysers for Tamil such as [16], [17], [21].

The ThamizhiFST is developed as a web based system and it has three layers. It consists of a preprocessing layer, a data layer and a layer of post-processing. The preprocessing layer and post processing layers are implemented using a server side web scripting language called PHP. The Data layer, or the FST layer, is implemented using a tool called FOMA [25].

1) Preprocessing layer: In the preprocessing phase, the text inputs are cleaned and normalised. Since the text in the morphological analyser is handled in Unicode form, it is important to normalise the text. In Tamil, the same character can be formed by multiple code sequences if it is not controlled or handled by the input method. For instance, the letter கொ sometimes can be input by the users using the following combinations: க + ொ or க + ெ + ா . However, this would lead to issues when executing regular expressions in the data layer. Therefore, it is important to normalise the text.

Further, Tamil words take a Sandhi marker when it precedes certain words due to phonological reasons. For instance, in the following sequence நடத்தப் பாேவதை the last letter of the first word ப் is called a Sandhi marker. In the preprocessing, these Sandhi letters are removed as they do not form part of the morphology.

2) Data layer: FOMA: This layer is the core part of the system where the actual morphological and orthographical rules are written. There are several tools available to implement the solutions including XFST [5], FOMA [25], OpenFST [2], and HFST [3]. One can also develop one's own FST from scratch. XFST is a propriety software which is available for academic research purposes. This is also widely used for grammar engineering using LFG and XLE [5], [23].

ThamizhiFST is implemented using the tool FOMA. FOMA complies with XFST and it can be integrated to the LFG that is written using XLE. In addition, FOMA has inbuilt support for the Unicode processing and rendering; more importantly this is an open source software that is available for anyone to download, modify and use.

The following steps have been used to develop an FST for the Tamil verbs:

- First the verbs are compiled and classified as described
- Next, as per the FOMA specification, rules were written for each class of verbs. A sample simple, rule is illustrated by the following:
  *% +verb% %|% +pres% %=*கிறு*% %|% +2sg% %=*ஆய்*:* கிறாய் *#.*
  As shown, it is not only the morpheme labels but also the corresponding morphemes which are also included in the rules. The % sign is used to escape the special characters and signs. A pipeline symbol is used as a delimiter to demarcate morphemes for ease of processing. The # is used to mark the lemma boundary. The idea of identifying morpheme boundaries is due to the inspiration from the term boundary introduced by [5].
- Finally, all the classes of rules were integrated and post-processed using rewrite rules. These rewrite rules are used to change the format of the FST output as required. For instance, though the ThamizhiFST is developed with all the labels and corresponding morpheme information, a set of rewrite rules were written to remove the morphemes (not the morpheme labels) and keep only label information when generating the FST for grammar engineering. This is because the XLE grammar needs only the morpheme labels, otherwise the tokenisation becomes unnecessarily complex.

3) Post-processing: In the post processing layer the output is converted to human readable form. This is because in the data layer several delimiters and notations are used to store morphemes and morpheme labels. In this layer this information is extracted, processed and displayed in a human readable form. Further, features like the Application Programming Interface for morphological analysis and generation, and an option to export the

[2]http://www.openfst.org
[3]https://github.com/hfst/

TABLE II
EVALUATIONS

Parameter	Value
TP	82
FP	2
FN	3
Precision	.976
Recall	.965
F-Measure	.970

results in form of UniMorph [22] will be incorporated in near future.

V. Evaluation and Discussion

A corpus of news text was compiled from a popular online Tamil news site that comprises all the genre of text such as politics, cinema, technology, social, business *etc.*

First, a testing dataset of 100 simple verbs were randomly picked from the news corpus, and then the morpheme and morphemes boundaries were identified manually. Next the system was evaluated and finally the harmonic mean of precision and recall called f-measure was calculated as per the formulas shown in (1), (2), and (3). This evaluation methodology was adapted from [21].

$$precision(P) = TP/(TP + FP) \tag{1}$$

$$recall(R) = TP/(TP + FN) \tag{2}$$

$$f - measure = 2 * P * R/(P + R) \tag{3}$$

where, TP = True Positives, FP = False Positives and FN = False Negatives. The values obtained during the evaluations are shown in the Table II.

The reason for FP and FN were due to issues in the rewrite rules. The words that were not recognised by the ThamizhiFST can easily be added to the system to improve the performance. The accuracy of ThamizhiFST can be improved by using a training set or a fine tuning set. In that way can find and correct the issues in the rewrite rules so that can reduce the FP and FN.

New words can be easily incorporated to the ThamizhiFST. If a new verb is found, then one needs to identify the verb class it belongs to based on its morphological features. Once the class is identified, one can simply add the verb to the corresponding class. There is no need for any further changes in the rule base. However, if a verb does not fit into the main 12 classes, then it can be added to the 13^{th} class and rules for this need to be added. In the future, coverage of compound verbs will be integrated into ThamizhiFST. A compound verb can be formed as in the formula shown in Formula 4:

$$Compoundverb = word^{\text{n}} + verb \tag{4}$$

where n≥1 and word = {noun, verb, adjective, *etc*}.

Further, like every other MAG author and developer, the ThamizhiFST has also been tested using its own development dataset. There are no benchmark data sets available for NLP tools in Tamil. As part of future work, together with other researchers within Tamil NLP, we hope to develop a benchmark dataset for NLP applications, including one for morphological analysis and generation.

VI. Related work

There are a number of studies which have been done on Morphological Analysers and / or Generators for South Asian Languages. Some of the relevant attempts are discussed in this section. A reference [26] proposes a MAG tool for the Nepali language. A classes of noun, pronoun, verbs, adjectives, numerals, adverbs, conjunction, postpositions and particles are identified, and then implemented as a pilot MAG for Nepali using the two level morphology and the FST approaches using XFST [26]. Another researcher [27] has developed a MAG for Sindhi as a part of his work on developing a Computational Grammar for Sindhi. The researcher has also used XFST to develop the MAG for Sindhi and then has integrated it to the LFG that is written using XLE.

A survey of computational morphology of Indian Languages has been carried out [28] and it documents 17 efforts on Morphological Analysis and/or Generator for the Tamil language. 12 of them were carried out before 2007. However, the papers, datasets and software are not retrievable via the Internet or databases. The rest were carried out in the year 2010. Among these 5 efforts [16], [17] and [21] are available for download in the binary form. However, no source code or data sets are available for reuse. Among these [21] and [16] have used only rule based approaches for the morphological generation. Further, [17] has applied a machine learning approach for the morphological analysis and generation of Tamil. It is claimed that the system was tested using 40,000 verbs and 30,000 nouns, and the machine learning system was trained using 130,000 verbs and 70,000 nouns from their own corpus [17]. However, the data sets or sources are not available except for a sample corpus with 270,000 tokens. A team has implemented a morphological analyser and generator for Tamil words [18], including verbs, nouns, adjectives, pronouns, numerals and non-standard Tamil words, using the toolbox of the Apertium tool. They have also evaluated the system using the CALTs and EMILLI corpora, and obtained 84% accuracy. However, unfortunately, the authors are not contactable. Another team has proposed a MAG for Tamil and implemented it using XFST in 2014 [29]. The authors have used transliteration to handle Tamil words. They have considered only 2000 noun and 96 verb stems for the analysis and generation. They have tested the system using their own data set consisting 3500 nouns and 500 verbs with a success of 78%.

VII. Conclusion

Our evaluation of the ThamizhiFST resulted in an f-measure of 0.97. This is a very good score for a language like Tamil that has very rich morphology. Further, since it was developed using a Finite State Transducer, the output can be easily changed and adapted to different needs. The ThamizhiFST can now analyse more than 800,000 surface forms of Tamil verbs. The coverage can be easily increased by just adding new verbs to the respective class files. A pilot version of ThamizhiFST can be accessed via parsers.projects.uom.lk.

A necessity for the incorporation of machine learning approach is identified and in future it will be used to improve the robustness of ThamizhiFST. Further, the present system will also be extended to cover complex verbs and other part of speech elements such as nouns. The proposed system will be further improved by integrating an Application Programming Interface (API) so that the service can be used by other applications easily.

Acknowledgement

The authors would like to thank Lauri Karttunen from Stanford University and Mans Hulden from the University of Colorado Boulder for their thoughts and technical support in making this work possible.

References

[1] L. H. George, "Statement on the status of tamil as a classical language," 2000, accessed on: 2017-11-12. [Online]. Available: https://southasia.berkeley.edu/tamil-classes

[2] K. Koskenniemi, "Two-level morphology," Ph.D. dissertation, University of Helsinki, 1983.

[3] T. Bögel, M. Butt, A. Hautli, and S. Sulger, "Developing a Finite-State Morphological Analyzer for Urdu and Hindi," in *Finite-State Methods and Natural Language Processing : Revised Papers of the Sixth International Workshop on Finite-State Methods and Natural Language Processing*, T. Hanneforth and K.-M. Würzner, Eds. Potsdam University Press, 2007, pp. 86–96.

[4] T. Bögel, "Urdu-Roman Transliteration via Finite State Transducers," in *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing (FSMNLP)*, 2012, pp. 25–29.

[5] K. R. Beesley and L. Karttunen, *Finite-state morphology*. Stanford: CSLI Publications, 2003.

[6] J. H. Martin and D. Jurafsky, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall, 2009.

[7] E. Annamalai, A. Dhamotharan, and A. Ramakrishnan, *Akarātiyi putiya patippil takālat tami ilakkaṇa viḷakkam*. Crea-A Publishers, 2014, pp. xxxi–xlvii.

[8] K. Paramasivam, *Contemporary Tamil Grammar*. Adaiyaalam, 2011.

[9] S. Agesthialingom, "A Note on Tamil Verbs," *Anthropological Linguistics*, pp. 121–125, 1971.

[10] L. Lisker, "Tamil verb classification," *Journal of the American Oriental Society*, vol. 71, no. 2, pp. 111–114, 1951.

[11] K. Graul, *Outline of Tamil grammar*. Leipzip University, 1855.

[12] A. H. Arden, *A progressive grammar of common Tamil*. Christian Literature Society, 1962.

[13] S. Rajaram, *English-Tamil Pedagogical Dictionary*. Thanjavur Tamil University, 1986.

[14] M. Irākavaiyaṅkār, *'Viaittiripu viḷakkam' (conjugation of Tamil verbs) (in Tamil)*. Eighty year anniversary publication, 1958.

[15] H. Sithiraputhiran, *Viaittiripu viḷakkamum moiyiyal kōtpāṭum*. International Institute of Tamil Studies, 2004.

[16] M. Anand Kumar, V. Dhanalakshmi, R. Rekha, K. Soman, and S. Rajendran, "A novel data driven algorithm for Tamil morphological generator," *International Journal of Computer Applications*, no. 12, pp. 52–56, 2010.

[17] M. Anand Kumar, V. Dhanalakshmi, K. Soman, and S. Rajendran, "A sequence labeling approach to morphological analyzer for Tamil language," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 2, no. 06, pp. 1944–195, 2010.

[18] K. Parameshwari, "An implementation of APERTIUM morphological analyzer and generator for Tamil," *Parsing in Indian Languages*, p. 41, 2011.

[19] S. Sulger, M. Butt, T. H. King, P. Meurer, T. Laczkó, G. Rákosi, C. B. Dione, H. Dyvik, V. Rosén, and K. De Smedt, "Pargrambank: The pargram parallel treebank," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 550–560.

[20] in *Cre-A: Dictionary of Contemporary Tamil*. India: Cre-A publishers, 2008.

[21] S. Menaka, V. S. Ram, and S. L. Devi, "Morphological generator for Tamil," *Proceedings of the Knowledge Sharing event on Morphological Analysers and Generators (March 22-23, 2010)*, pp. 82–96, 2010.

[22] C. Kirov, J. Sylak-Glassman, R. Que, and D. Yarowsky, "Very-large scale parsing and normalization of wiktionary morphological paradigms," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, N. C. C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Paris, France: European Language Resources Association (ELRA), may 2016.

[23] M. Butt, T. H. King, M.-E. Nino, and F. Segond, *A Grammar Writer's Cookbook*. Stanford: CSLI Publications, 1999.

[24] M. Nuhman, *Basic Tamil Grammar (In Tamil)*. Readers' Association, Sri Lanka, 1999.

[25] M. Hulden, "Foma: a finite-state compiler and library," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*. Association for Computational Linguistics, 2009, pp. 29–32.

[26] B. Prasain, "Computational analysis of nepali morphology: A model for natural language processing," Ph.D. dissertation, Faculty of Humanities and Social Sciences of Tribhuvan University, Nepal, 2011.

[27] M. U. Rahman, "Developing a sindhi computational resource grammar in lexical functional grammar framework," Ph.D. dissertation, Faculty of Engineering Science and Technology, Isra University, Hyderabad, 2016.

[28] P. Antony and K. Soman, "Computational morphology and natural language parsing for Indian languages: a literature survey," *International Journal of Scientific and Engineering Research*, vol. 3, 2012.

[29] S. Lushanthan, A. Weerasinghe, and D. Herath, "Morphological analyzer and generator for Tamil language," in *Advances in ICT for Emerging Regions (ICTer), 2014 International Conference on*. IEEE, 2014, pp. 190–196.