Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

A Hybrid Approach to Myanmar Morphological Analysis and Generation Using Finite State Techniques and LSTM-Based Deep Learning Models

Kaung Myat Thu, H. Mamata Devi, Th. Rupachandra Singh

Department of Computer Science, Manipur University, Imphal, India
Email: kaungmyatthu.kmt@gmail.com

This study presents a hybrid approach to Myanmar morphological analysis and generation (MAG) by combining Finite State Techniques (FSTs) with Long Short-Term Memory (LSTM) models. Myanmar's complex morphological structures, such as affixation and morphotactics, and the lack of linguistic resources pose significant challenges. FSTs provide a rule-based framework for morphotactics, while LSTMs enhance adaptability by learning sequential patterns from data.
We created MM-Morph-Shallow and MM-Morph-Deep datasets, containing over 11 million tagged word pairs, and evaluated the system using these datasets. Experimental results show high accuracy, with MM-Morph-Shallow achieving 98.98% accuracy for verbs and 98.70% for nouns, while MM-Morph-Deep achieves 97.98% and 97.70%, respectively. A user-friendly GUI, MM-Morph (v2), was developed to facilitate real-time morphological analysis, enabling researchers and linguists to interact with the system efficiently. This research advances Myanmar NLP, enabling machine translation, spell-checking, grammar-checking and text generation applications.

Keywords: Myanmar language, Morphological Analysis and Generation (MAG), Finite State Techniques, LSTM, Hybrid System, NLP, XFST, LEXC.

1. Introduction

Morphological analysis and generation are vital components of computational linguistics, playing a key role in understanding the structure and semantics of words. These tasks involve breaking down words into their morphemes (the smallest meaning-bearing units) and generating word forms based on morphological rules. Morphological processing enables deeper linguistic insights and facilitates the development of robust NLP applications across languages.

The Myanmar language poses several challenges due to its complex word structures. It features extensive use of affixes, reduplication, and compounding. The lack of linguistic resources, annotated datasets, and standardized tools for Myanmar language processing further compounds these challenges.

To address these challenges, this research combines two approaches: Finite State Techniques (FSTs) and Long Short-Term Memory (LSTM) networks. Finite State Techniques are used as the primary method to define and implement Myanmar's morphological rules. However, FSTs alone struggle with exceptions, ambiguous patterns, and unseen word forms.

To overcome these limitations, LSTM-based deep learning models are integrated into the system. LSTMs are well-suited for processing sequential data and can learn complex patterns from large datasets. By combining FSTs with LSTM models, this study introduces a hybrid system that blends rule-based precision with the adaptability of machine learning. The FSTs provide a strong foundation for modeling the structure of Myanmar words, while LSTMs improve the system's accuracy by learning from data and handling variations.

2. Literature Review

The table below compares various methods from the literature, providing the rationale for selecting a hybrid approach that combines Finite State Techniques (FSTs) with deep learning models like LSTM networks.

Table 1: Comparative overviews of related works

Method	Description	Applicability to Myanmar Language	Ref No
Corpus-Based ML	Learns patterns from large datasets using ML models like HMMs and RNNs.	Limited by Myanmar's scarce linguistic resources and complex morphology.	[1],[2],[3],[4], [5],[6] [7],[8],[9]
Finite State Techniques (FSTs)	Uses finite state automata for precise rule-based modeling.	Effective for Myanmar's rule-based morphology but needs integration with adaptive methods for exceptions.	[10],[11],[12], [13], [14],[15] [16],[17],[18]
Hybrid FST + Deep Learning	Combines FSTs with adaptive models like LSTM for enhanced accuracy.	Best suited for Myanmar, addressing both rule-based and probabilistic variations.	[19],[20],[21], [22], [23],[24] [25],[26],[27]
Paradigm-Based Approach	Uses templates of word inflection patterns (paradigms) to analyze and generate word forms.	Efficient for Myanmar's agglutinative morphology but limited by the need for extensive paradigm libraries.	[28],[29],[30]
Suffix Stripping	Removes affixes to find root forms using predefined rules.	Struggles with Myanmar's irregular forms and nuanced affixation.	[31],[32],[33]
DAWG Approach	Represents word forms in graph structures for efficiency.	Useful for compactly representing variations but requires extensive pre-processing.	[34],[35],[36]

A thorough study shows that combining finite-state techniques with LSTM deep learning is highly effective for morphological analysis and generation. Finite-state techniques provide a strong mathematical foundation, while LSTMs enhance the system's ability to handle complex patterns. This conclusion is supported by a review of existing research, which highlights this hybrid approach as the most suitable solution for language processing tasks.

3. Methodology

The methodology is detailed in the following steps:

- Identification and Categorization of Morphemes
- Listing Morphemes and Lexical Categories
- Collection of Roots and Affixes

- Defining and Specifying Morphotactics
- Development of Finite State Transducers (FSTs)
- Dataset Preparation and Formatting
- Training LSTM Models
- Morphological Analysis of Word Classes
- Implementation of a Graphical User Interface (GUI)
- Testing and Evaluation

4. The Study of Myanmar Morphology

Morphology is a branch of linguistics that studies the structure and formation of words. It focuses on understanding how the smallest units of meaning, called morphemes, combine to create words with different meanings and grammatical roles. This study systematically analyzed the Myanmar language to identify and classify morphemes, such as roots, prefixes, and suffixes, which are essential for understanding its morphological structure. The analysis focused on categorizing morphemes into lexical categories like nouns, verbs, adjectives, and adverbs, establishing the foundational linguistic components required for computational processing. A detailed inventory of morphemes and their corresponding lexical categories was compiled to support lexicon development. This inventory includes both frequent and less frequent morphemes, ensuring broad coverage of Myanmar's rich morphological structures. The following table shows the morphological classification of Myanmar affixes throughout extensive linguistic studies.

Table 2. The morphological classification of Myanmar affixes

No	Affix names	Morphological tags	No	Affix names	Morphological tags
1	Subjunctive	SUBJ	48	Ablative	ABL
2	Imperative	IMP	49	Associative	ASSO
3	Suggestive	SUG	50	Connective	CON
4	Present	PREASP	51	Sequential Conditional	SECQ
5	Past	PAASP	52	Inquisitive	INQ
6	Perfect	PERASP	53	Interrogative	INTERQ
7	Continuative	PROGASP	54	Definitive	DEFI
8	Future	FUTASP	55	Subjunctive	SUBJUNC
9	Condition	COND	56	Emphatic	EMPH
10	Plural	PL	57	Distributive	DISTR
11	Experimental	EXP	58	Classifier	CLASS
12	Concessive	CONC	59	Matter	MAT
13	Copula	COP	60	Relation	REL
14	Priority	PRI	61	Limit	LIM
15	Compassion	COMP	62	Manner	MAM
16	Aspiring	ASP	63	Substitution	SUBT
17	Adverbial	ADVAL	64	Addition	ADD
18	Adjectival	ADJVAL	65	Realis	REAL
19	Nominalizer	NZRP	66	Generalization	GENR
20	Nominalizer	NZRS	67	Purpose	PURP
21	Subordinate	SUBO	68	Politeness	POL
22	Reason	REA	69	Benefactive	BEN
23	Purpose	PUR	70	Quotation	QUOT

24	Evidential	EVD	71	Reciprocal	RECIP
25	Accidental	ACCD	72	Subordinate	SUBORD
26	Un-Expectational	UNEXP	73	Honorific	HON
27	Expectational	EXP	74	Comparison	COMP
28	Just	JUST	75	Similarity	SIM
29	Doubt	DOBT	76	Measure	MSR
30	Habitual	HAB	77	Identity	IDEN
31	Endearment	DEAR	78	Topic	TOP
32	Precative	CAUS	79	Appellative	APPEL
33	Modality	MOD	80	Euphonic	EUPH
34	Probability	PRB	81	First Person	1P
35	Honorific	HON	82	Second Person	2P
36	Question Final Marker	Q-WH	83	Third Person	3P
37	Interrogative Marker	INTER	84	Q-Wh	Q-WH
38	Negative Marker	NEGP	85	Demonstrative	DEMO
39	Comparative	COMP	86	Distal	DIST
40	Causative	CAUS	87	Nominalizer	NZRP
41	Euphonic	EUP	88	Nominalizer	NZRS
42	Singular	SG	89	Nominative	NOM
43	Plural	PL	90	Instrumental	INSTR
44	Masculine	MASC	91	Accusative	ACC
45	Feminine	FEM	92	Dative	DAT
46	Causative	CAUS	93	Genitive/	GEN
47	Locative	LOC			

The figure below illustrates the key functions of morphological analysis and generation, using the example word "ကလေးမများအတွက်ကြောင့်ပါ." The word is analyzed and broken down into its root word and corresponding grammatical morphemes: ကလေး + NOUN + FEM + PL + BEN + CAUS + POL.

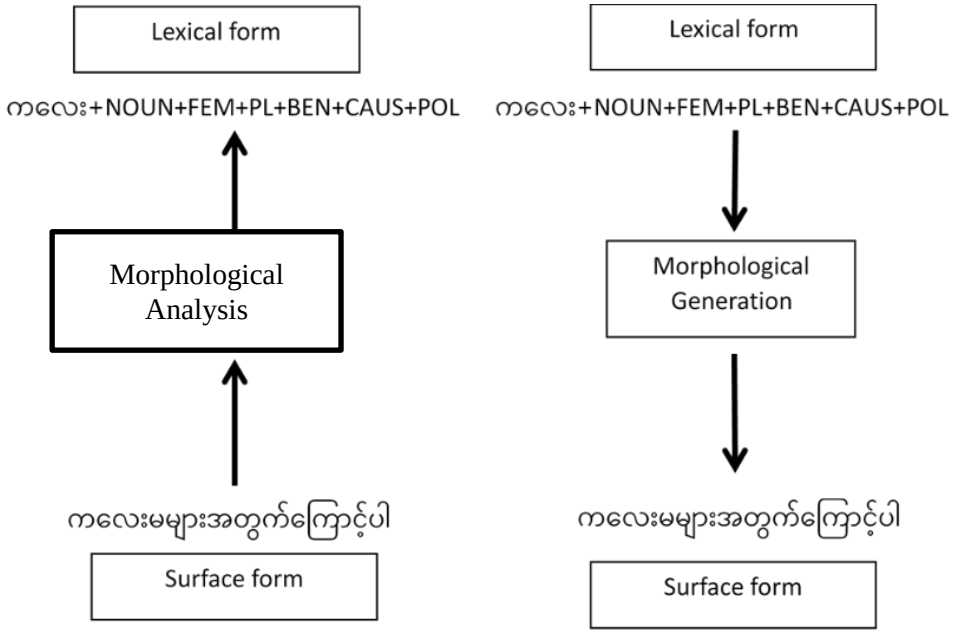

Figure 1: The key functions of morphological analysis and generation

5. Theoretical Framework for FST and LSTM Deep Learning Models

The figure below illustrates the proposed taxonomy for developing the Morphological Analysis and Generation (MAG) system.

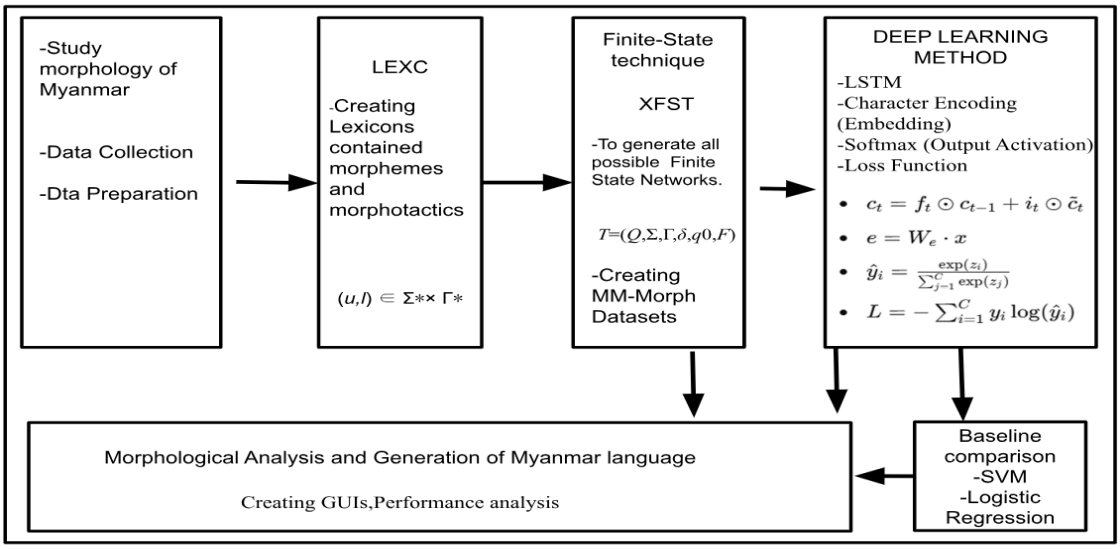

Figure 2: The proposed taxonomy for developing the Morphological Analysis and Generation (MAG) system.

The theoretical framework of this study is divided into two parts.

- Finite State Techniques (FSTs) focus on their role in defining morphotactics and generating surface-lexical word pairs as input data. Finite-state Transducers (FST) A finite-state transducer can be represented as a 6-tuple, T=(Q, Σ, Γ,δ,q0, F), where: Q is a finite set of states, Σ is the input alphabet, Γ is the output alphabet, δ is the transition function $Q\times(\Sigma\cup\{\varepsilon\})\times(\Gamma\cup\{\varepsilon\})\rightarrow Q\times\Gamma*$, q0 is the initial state, F is the set of final states.

- Long Short-Term Memory (LSTM) models are used to enhance the system's ability to learn and predict complex morphological patterns. LSTMs consist of repeating modules, each containing three main gates: forget, input, and output. These gates regulate the flow of information through the network, allowing it to retain or discard information as needed. Input characters to numerical representations using embeddings or one-hot encoding. These embeddings are then processed through the layers of the network

Forget Gate: Determines what information to discard from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate: Determines what new information to add to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cell State Update: Combines the forget and input gates to update the cell state

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}$$

Output Gate: Determines the hidden state for the next time step.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

Each character in the sequence is processed in order. The hidden state h_t captures dependencies between characters and morphemes, while the cell state C_t retains long-term information.

5.1 LSTM Model Step-by-Step Example with The Burmese Word Sequence

Burmese word sequence: "ကလေးမများအတွက်ကြောင့်ပါ

Words in Sequence: "ကလေး" → "မ" → "များ" → "အတွက်" → "ကြောင့်" → "ပါ". LSTM model to remember that "ကလေး" is the noun root, and track affixes like "များ" (plural) and "အတွက်" (beneficiary).

- Time Step 1: Input "ကလေး"

○ Forget Gate f_t : Decides what to forget from the cell $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$ (initially zero).

○ Input Gate i_t : Decides what to add (e.g., "NOUN"). $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

○ "ကလေး" as NOUN and updates the memory C_t

- Time Step 2: Input "မ"

○ Forget Gate f_t : Decides if "NOUN" information should persist.

○ Input Gate i_t : Adds new information for "FEM" (feminine affix).

○ Cell State Update C_t : Memory now stores "NOUN + FEM".

- Time Step 3: Input "များ"

○ Forget Gate f_t: Keeps previous memory (NOUN + FEM).

○ Input Gate i_t : Adds "PL" (plural suffix).

○ Cell State C_t : Updates to "NOUN + FEM + PL".

- Time Step 4–6: Input "အတွက်", "ကြောင့်", "ပါ"

Adds BEN (beneficiary marker), CAUS (causative marker), and POL (politeness marker).

Final Memory C_t: Contains "NOUN + FEM + PL + BEN + CAUS + POL". In this example,

LSTMs avoid forgetting earlier steps (e.g., "ကဖလး") due to the cell state C_t . The forget gate allows the model to control which parts of the memory are important.

6. Dataset Creation Processes

The creation of datasets for Burmese morphological analysis followed six key steps:

1. Data Collection: Data were gathered from sources like mypos-ver.3.0 corpus and English-Myanmar Dictionary databases and grammar books

2. Data Preparation: Categorization of roots and affixes and morphological tagging ensured compatibility with computational tools.

3. Lexicon Construction: Shallow and deep morphotactic lexicons were built using the LEXC tool to represent varying levels of morphological detail.

4. Finite-State Compilation: Lexicons were compiled into finite-state transducers (FSTs) for efficient morphotactic modeling.

5. Word Pair Generation: Surface-lexical word pairs were generated using the XFST tool, linking surface forms to their lexical representations.

6. Dataset Creation: Generated pairs were refined into MM-Morph-Shallow and MM-Morph-Deep datasets, incorporating detailed morphological tags for machine learning applications. MM-Morph-Shallow and MM-Morph-Deep datasets comprise over 11 million morphologically tagged word pairs.

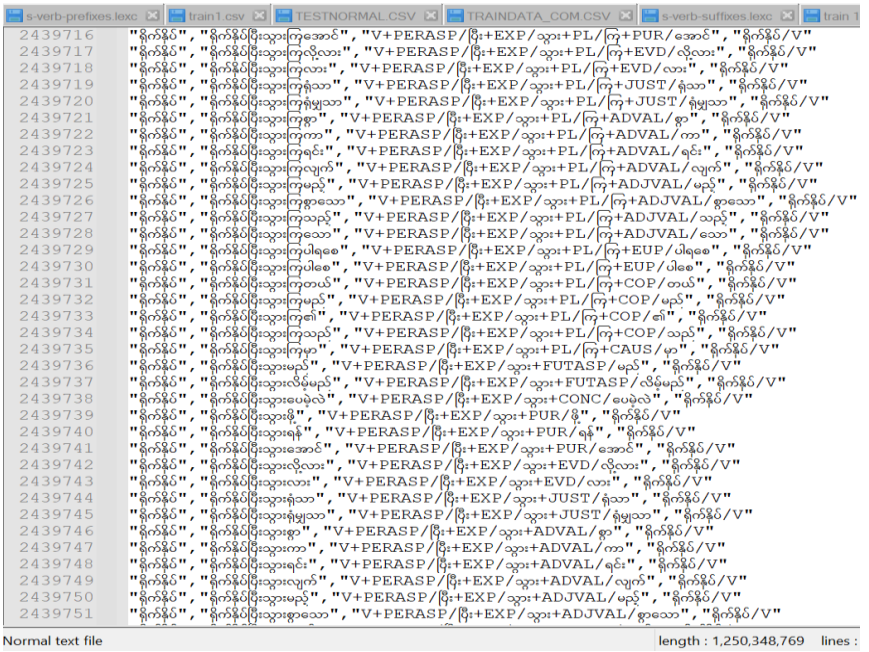

Figure 3: Sample of the morphologically tagged dataset by refining the output of FSTs.

The dataset captures various linguistic features, such as tense, number, and case, ensuring the system can handle a wide range of morphological phenomena. The uses of tools like XFST, LEXC, and the creation of lexicons are detailed in our previous chapters [41][42].

7. Evaluation and Analysis

The lexicon was divided into training, validation, and testing subsets to train the LSTM-based model. The training set comprised 80% of the dataset, including frequent and infrequent morphological patterns, to ensure robust learning. The validation set (10%) was used for hyperparameter tuning, while the test set (10%) contained unseen examples to evaluate the model's generalizability.

During the training process, surface forms were tokenized at the character level to represent the granular structure of the Myanmar language. Each character was encoded using one-hot encoding or embedding layers to convert them into numerical representations, which are processed as input sequences. This step is crucial as character-level tokenization allows the model to capture intricate morphological patterns, such as affixation and morphotactics which are common in Myanmar words. The LSTM model processes these input sequences by learning the dependencies between characters and their corresponding morphological labels. The model optimizes its parameters through backpropagation using a predefined loss function (e.g., categorical cross-entropy), which measures the difference between predicted and actual labels. The Adam optimizer iteratively adjusts the weights to minimize the loss and improve predictions.

We evaluated the performance of the LSTM model using metrics such as accuracy, precision, recall, and F1-score. These metrics provide quantitative measures of how well the system is performing and help identify areas for improvement. During training, the model adapts to both frequent and rare patterns by generalizing from the examples, enabling it to accurately predict morphological features, even for unseen word forms in the test set. This combination of character-level encoding, robust training processes, and iterative optimization ensures the model's ability to capture Myanmar's complex morphological structures effectively.

Table 3: A performance results for LSTM models on MM-Morph-Deep and MM-Morph-Shallow datasets

Type	Word pairs in a Datasets	Precision	Recall	F1 Score	Accuracy
MM-Morph-Deep					
Nouns	3908171	0.97	0.976	0.973	97.7
Verbs	6371670	0.9742	0.9808	0.9775	97.98
MM-Morph-Shallow					
Nouns	423150	0.9852	0.986	0.9856	98.7
Verbs	423150	0.9842	0.9808	0.9825	98.98

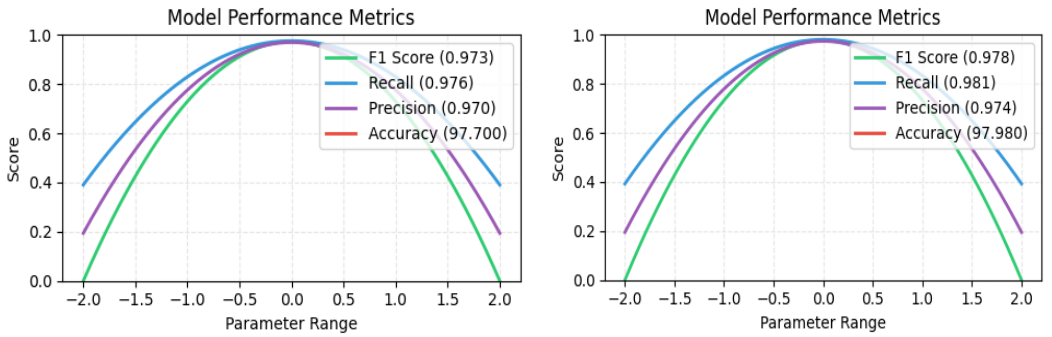

Figure 4,5: Showing the model's performance curves for evaluation of nominal and verbal word forms in the MM-Morph-Deep dataset

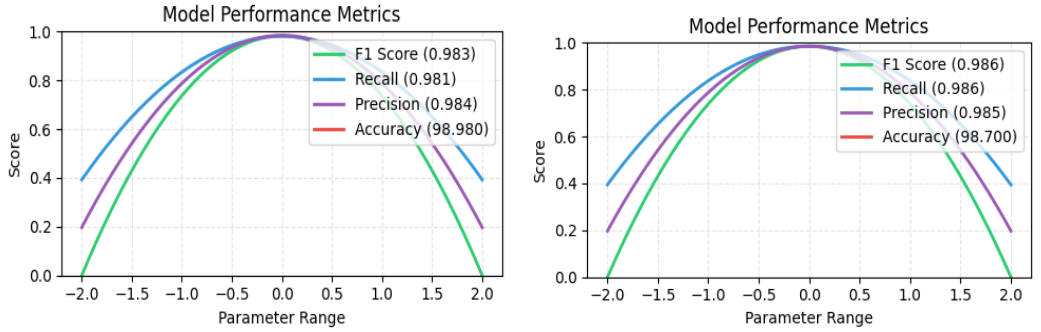

Figure 6,7: Showing the model's performance curves for evaluation of nominal and verbal word forms in the MM-Morph-Shallow dataset

Examples of Correct Predictions

These examples demonstrate the system's ability to predict morphological patterns in the test words. The system output:

✓ CORRECT PREDICTION

Derived Word: သွဝါဒခံကြရအောင်

True Root: သွဝါဒခံ

True Label: V+PL/ကြ+UNEXP/ရ+PUR/အောင်

Predicted Label: V+PL/ကြ+UNEXP/ရ+PUR/အောင်

Label Meanings:

+V/သွဝါဒခံ word: Verb

+PL/ကြ: Plural

+UNEXP/ရ: Un-Expectational

+PUR/အောင်: Purpose

Examples of Errors

The typical error comes from classifying infrequent tags like +ADVAL. Analysis revealed that insufficient training examples for complex morphological structures often caused these errors.

The system output:

✗ WRONG PREDICTION

Derived Word: နေထိုင်ကြရလျက်

True Root: နေထိုင်

True Label: V+PL/ကြ+UNEXP/ရ+ADVAL/လျက်

Predicted Label: V+PL/ကြ+UNEXP/ရ+COMP/ရှာ+ADVAL/လျက်

Expected: V+PL/ကြ+UNEXP/ရ+ADVAL/လျက်

Got: V+PL/ကြ+UNEXP/ရ+COMP/ရှာ+ADVAL/လျက်

Label Meanings:

+V/သဝါဒခံ word: Verb

+PL/ကြ: Plural

+UNEXP/ရ: Un-Expectational

+COMP/ရှာ: Compassion

+ADVAL/လျက်: Adverbial

Error Analysis

Errors were categorized into the following types:

1. Label ambiguity: Confusion between similar tags (e.g., +NZR+SG vs. +NZR+PL).
2. Morphological complexity: Difficulty in handling multi-layered morpheme structures.
3. Out-of-Vocabulary (OOV): Limited ability to generalize to rare or unseen root words.

There are several steps to improve the model's performance, such as implementing data augmentation and refining the character-level tokenization process.

8. Implementation of GUI

We developed a tool called MM-Morph (v2), which features a graphical user interface (GUI) to streamline the entire workflow. Built using Python and its machine learning libraries, this application provides an accessible, easy-to-use interface to engage with the various components of the analysis system. Through MM-Morph (v2), researchers and linguists can perform essential tasks such as training models, testing datasets, evaluating system performance, and exporting analyzed data for further use. The tool also has a real-time evaluation feature, enabling users to instantly input word forms and receive morphological analysis results.

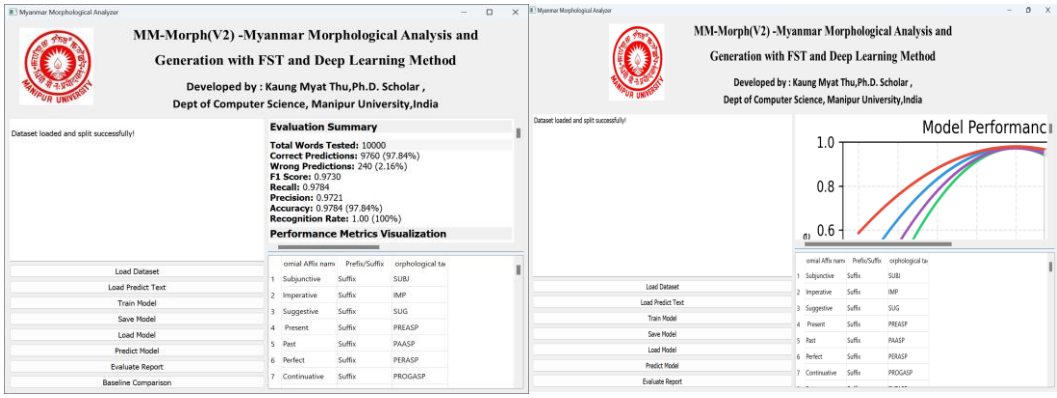

Figure 8,9: Showing evaluation report in MM-Morph(v2)

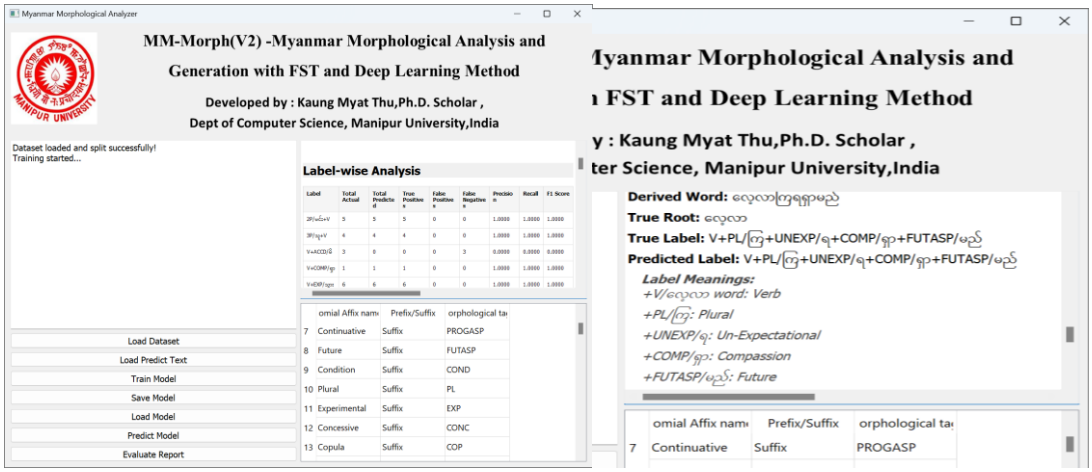

Figure 10: Label-wise analysis / Figure 11: Morphological analysis in MM-Morph(v2)

9. Conclusion

Through this research, we have classified nominal and verbal suffixes and assigned linguistic terms for these affixes for the first time in the history of Myanmar linguistic studies. We analyzed their patterns and behaviors in word formation, enabling us to study Myanmar's linguistic phenomena. While we could not cover all the complexities of Myanmar's morphotactics, this work provides a foundation for future linguistic research to expand.

The findings and tools developed from this research have applications beyond linguistic analysis. They are ready to use for language learning, spell-checking, text generation, machine translation, and other NLP tasks. This research contributes to preserving and studying Myanmar's linguistic heritage, advancing technology development, and bridging gaps in academic resources.

References

1. Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. Computational Linguistics, 27(2), 153–198.
2. Goldsmith, J. (2006). An algorithm for the unsupervised learning of morphology. Natural Language Engineering, 12(4), 353–371. https://doi.org/10.1017/S1351324905004055
3. Welgama, V., Weerasinghe, R., & Niranjan, M. (2013). Evaluating a machine learning approach to Sinhala morphological analysis. In Proceedings of the 10th International Conference on Natural Language Processing (pp. 1–8). Noida, India.
4. Lee, J. (2008). A nearest-neighbor approach to the automatic analysis of Ancient Greek morphology. In Proceedings of CoNLL 2008: 12th Conference on Computational Natural Language Learning (pp. 127–134).
5. Kumar, M. A., Dhanalakshmi, V., Soman, K. P., & Rajendran, S. (2010). A sequence-labeling approach to morphological analyzer for Tamil language. International Journal on Computer Science and Engineering, 2(6), 2201–2208.
6. Kruengkrai, C., Sornlertlamvanich, V., & Isahara, H. (2006). A conditional random field framework for Thai morphological analysis. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC). Genoa, Italy.
7. Jedrzejowicz, P., & Strychowski, J. (2005). A neural network-based morphological analyzer of the natural language. In Text, Speech and Dialogue: 7th International Conference (pp. 199–208). Springer. https://doi.org/10.1007/3-540-32392-9_21
8. Premjith, B., Soman, K. P., & Kumar, M. A. (2018). A deep learning approach for Malayalam morphological analysis at character level. Procedia Computer Science, 132, 47–54.
9. Abate, M., & Assabie, Y. (2014). Development of Amharic morphological analyzer using memory-based learning. In Advances in Natural Language Processing (pp. 1–13). Springer.
10. Katushemererwe, F., & Hanneforth, T. (2010). Finite state methods in morphological analysis of Runyakitara verbs. Nordic Journal of African Studies, 19(1), 22–32.
11. Kayabaş, A., Schmid, H., Topçu, A., & Kılıç, Ö. (2019). TRMOR: A finite-state-based morphological analyzer for Turkish. Turkish Journal of Electrical Engineering & Computer Sciences, 27(5), 3837–3851.
12. Beesley, K. R., & Karttunen, L. (2003). Finite-state morphology: Xerox tools and techniques. CSLI Publications.
13. Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., & Mohri, M. (2007). OpenFST: A general and efficient weighted finite-state transducer library. In Implementation and Application of

Automata (pp. 11–23).
14. Lindén, K., Silfverberg, M., & Pirinen, T. (2009). HFST tools for morphology: An efficient open-source package for construction of morphological analyzers. In International Workshop on Systems and Frameworks for Computational Morphology (pp. 28–47).
15. Hulden, M. (2009). 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 (pp. 29–32).
16. Abebe, T., Washington, J., Gasser, M., & Yimam, B. (2018). A finite-state morphological analyzer for Wolaytta. In Information and Communication Technology for Development for Africa (pp. 18–26). Springer. https://doi.org/10.1007/978-3-319-95153-9_2
17. Zueva, A., Kuznetsova, A., & Tyers, F. M. (2020). A finite-state morphological analyser for Evenki. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC) (pp. 2581–2589).
18. Sarveswaran, K., Dias, G., & Butt, M. (2018). ThamizhiFST: A morphological analyser and generator for Tamil verbs. In 2018 3rd International Conference on Information Technology Research (ICITR) (pp. 1–6).
19. Premjith, B., Soman, K. P., & Kumar, M. A. (2018). A deep learning approach for Malayalam morphological analysis at character level. Procedia Computer Science, 132, 47–54.
20. Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN) (pp. 2047–2052).
21. Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research, 3, 115–143.
22. Hajič, J., Hlaváčová, J., & Hladká, B. (2018). Morphological analysis of Czech: Current challenges. In Proceedings of the International Conference on Computational Linguistics (COLING) (pp. 205–212).
23. Pirinen, T. A. (2019). Finite-state morphological analysers for Northern Sámi and other Uralic languages. Language Documentation & Conservation, 13, 207–227.
24. Oflazer, K. (1994). Two-level description of Turkish morphology. Literary and Linguistic Computing, 9(2), 137–148.
25. Washington, J. N., & Gasser, M. (2018). A finite-state morphological analyzer for Wolaytta. Proceedings of the 27th International Conference on Computational Linguistics (COLING), 2422–2432.
26. Washietl, S., & Kornai, A. (2020). Morphological analysis of Hungarian using HFST. Language Resources and Evaluation, 54(1), 1–13. https://doi.org/10.1007/s10579-019-09459-6
27. Antony, P. J., Kumar, M. A., & Soman, K. P. (2010). Paradigm-based morphological analyzer for Kannada language using machine learning approach. Advances in Computer Science and Technology, 3, 457–481.
28. Shah, D. N., & Bhadka, H. (2020). Paradigm-based morphological analyzer for the Gujarati language. Advances in Intelligent Systems and Computing, 989, 501–509. https://doi.org/10.1007/978-981-13-8618-3_50
29. Durrett, G., & DeNero, J. (2013). Supervised learning of complete morphological paradigms. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 1185–1195. http://nlp.cs.berkeley.edu
30. Porter, M. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137. https://doi.org/10.1108/eb046814
31. Lovins, J. B. (1968). Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11(1), 22–31.
32. Paice, C. D. (1990). Another stemmer. SIGIR Forum, 24(3), 56–61.

https://doi.org/10.1145/101306.101310
33. Maynard, A., et al. (2006). Efficient representation of morphological variations in Turkish using DAWGs. Journal of Computational Linguistics, 32(4), 451–468.
34. Janicki, P., et al. (2014). DAWG models for inflectional morphology in Slavic languages. Transactions of the ACL, 2, 243–256.
35. Aho, T., & Jääskeläinen, M. (2017). Multilingual dictionary management using directed acyclic word graphs. In Computational Lexicography Conference Proceedings (pp. 87–98).
36. Maung, T. K. (1983). Structure and usage of affixes in Burmese. Journal of Myanmar Linguistics, 15(3), 120–132.
37. Aye, M. (2008). The soul of Myanmar: Language and identity. Myanmar Studies Journal.
38. Phyue, S. L., & Thida, A. (2012). Finite-state morphology for Burmese: Challenges and implementation. In Proceedings of the International Conference on Computational Morphology.
39. Hla Hla Htay, G., Bharadwaja Kumar, & Murthy, K. N. (2007). Statistical analyses of Myanmar corpora. Technical Report. Department of Computer and Information Sciences, University of Hyderabad.
40. Hlaing, T. S. (2008). The syntax of focus and topic in Burmese. In Proceedings of the Myanmar Linguistic Symposium (pp. 45–55).
41. Thu, K. M., Devi, H. M., & Singh, T. R. (2024). A computational implementation of morphological analysis and generation of verbs in Myanmar language. International Conference on Data Science, AI and Analytics: Bridging the Gap Between Theory and Practices (ICDSAIA-2023), Taylor's University, Malaysia. International Journal of Intelligent Systems and Applications in Engineering (IJISAE), 12(8s), 615–622. https://doi.org/
42. Thu, K. M., Devi, H. M., & Singh, T. R. (2023). MM-Morph: A computational linguistic tool for morphological analysis and generation of Myanmar nouns. 4th International Conference on Multidisciplinary and Current Educational Research (ICMCER-2023), Thailand. International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), 10(5), 68. https://doi.org/