ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

Comparative Studies on Some Morphological Analysis and Generation Techniques for Myanmar Language: A Review

[1] Kaung Myat Thu, [2] H. Mamata Devi, [3] Th. Rupachandra Singh

[1] [2] [3] Department of Computer Science, Manipur University, Imphal, Manipur, India
Corresponding Author Email: [1] kaungmyatthu.kmt@gmail.com, [2] mamata_dh@rediffmail.com, [3] rupachandrath@gmail.com

Abstract*— Morphological Analysis and Generation (MAG) are essential in natural language processing, especially for morphologically rich languages. It is the first step toward every NLP task, including lemmatization, POS tagging, spell checking, grammar checking, machine translation, text summarization, and information extraction. MAG deals with the study of word formation and grammatical structure inside a word. Every MAG task comprises three main parts: a morpheme lexicon, a set of morphotactic or orthographic rules, and decision algorithms. This paper has reviewed some popular approaches that many researchers have taken. We found that the Corpus-based machine learning approach (SVM, NN, CRF, MDL, ...), Paradigm based approach, two-level technique, Finite State Automata (FSA) based techniques, Finite State Transducers (FST) based techniques, Suffix stripping, DAWG (Directed Acrylic Word Graph) are popular, successful methods reported in the literature. Few or no research and developments in morphological analysis and generation for the Myanmar language have made this study a review of the literature on other similar languages.

***Keywords*— Morphology, Natural Language Processing, FST, FSA, Morphological analysis, and generation (MAG), Indian Language, Myanmar Language, XEROX FST, OpenFST, HFST, FST, SFST, Aptertium.**

I. INTRODUCTION

The morphological Analysis and Generation (MAG) task is concerned with analyzing word formation, identifying grammatical structure within words, and generating words according to the morphotactic rules. Every word is composed of the smallest meaning-bearing components called morphemes. There are two main classes of morphemes called stems and affixes. The stems, also called free morphemes, are the "main" morphemes of the word, acting as the main meaning, while the bound morphemes, called the affixes which, support "additional" meanings of various kinds. The study on MAG is the very first step toward every NLP task and is widely used in NLP applications such as search engines, speech recognition, lemmatization, POS tagging, spell checking, grammar checking, machine translation, text summarization, and information extraction. The morphological analyzer and generator are two essential computational programs that can identify the stems, affixes, and grammatical structures of words and generate new words, i.e., their lexical forms and surface forms. The analyzer classifies the input words and returns the stem of those words associated with their grammatical information. The morphological generator does the reversible function of the analyzer, i.e., for a given root word and grammatical information, the program will generate the surface form of words.

The area of this study is a computational approach to linguistics and an effort to give the ability to understand linguistics to the computer system. According to the report, researchers have taken various methods to implement morphological analysis and generation for different languages. Few or lack of research and reliable development systems in MAG for the Myanmar language have made this study to make a review of the literature of other similar languages. Myanmar belongs to the Sino-Tibetan language family, which has a complex morphological structure. Because of its agglutinative nature and complexity in morphological structure, the computerization of morphological phenomena for the Myanmar language is quite challenging. Corpus-based machine learning approach (SVM, NN, CRF, MDL, ...), Paradigm-based approach, two-level technique, Finite State Automata (FSA) based techniques, Finite State Transducers (FST) based techniques, Suffix stripping, DAWG (Directed Acrylic Word Graph) are popular successful methods reported in the literature.

The structures of the rest of this paper are as follows. In section (2), we study existing approaches researchers have taken to implement morphological analysis and generation for different languages. In section (3), a comparison has been made among popular methods to determine the advantages and disadvantages. Finally, section (4) concludes the paper with a description of further works.

II. SOME STUDIES ON EXISTING APPROACHES

John Goldsmith (2001) [1] presented a Corpus-based approach or Statistical approach by using minimum description length algorithm (MDL) to develop unsupervised learning of morphology for many European languages. Corpora ranging in size from 5000 to 500000 words have

been used for this approach. Corpus is a large collection of texts which can be used to develop and evaluate NLP applications. In this approach, a raw corpus is supported as input to generate segments of words as an output. The morphological structure of words can be generated from those segmented words. Morpheme segmentation can also be performed using a statistical approach, which requires training on an excessive amount of data. Labeled, unlabeled, or partially labeled corpora have been used as the training data in this method. The accuracy depends on the size and type of the corpus. Statistical-based morphological analyzers can be performed either supervised or unsupervised using a different kind of corpus. The overall accuracy rates achieved were 82.9% for English and 83.3% for French, respectively.

John Goldsmith (2006) [2] developed a famous linguistic software called Linguistica, which was implemented based on the algorithm called minimum description length (MDL). The work described in his paper is part of a project aimed at the unsupervised learning of natural language morphology. It can be assumed as suffix-based morphology. Morphological analysis is performed based on the task of splitting a word into distinct, successive morphs rather than the allocation of morphosyntactic features. On the first 200,000 and the first 300,000 words of the Brown corpus, Linguistica achieved an accuracy of 72%.

Viraj Welgama et al. (2013) [3] reported an evaluation of a popular morph segmentation algorithm (Morfessor) for Sinhala Morphological analysis using a corpus-based machine learning approach. They used the SGSD (Sinhala Gold Standard definitions) version 1.0, which contains the morphs for 435,076 unique Sinhala words. To train and test the Morfessor algorithm, tagged UCSC 10M Words Sinhala corpus and 69,735 words extracted from UCSC corpus are used. Morfessor, a morph segmentation algorithm, was developed by the organizers of the Morph challenge competition. This competition was introduced in 2005 to promote and compare their algorithms among the machine learning community, linguists, and specialists in NLP applications. Some agglutinative languages like Turkish and Finnish, which are more complex than European languages, were successful to some extent with the help of this competition. Morfessor takes a list of raw words as input with their corresponding frequencies and generates morpheme segments of the word forms. It is confirmed that the algorithm can correctly obtain 35% morphemes analyses and 56% accuracy in the identification of stems. The authors of Morfessor algorithms used F-measure-based evaluation to evaluate the performance of the system. According to reports, this method can be extended to perform complete morphological analysis.

John Lee (2008) [4] presented a data-driven method for the Morphological analysis of Ancient Greek. This system requires no man-made rules to classify morphological features automatically through the use of a nearest-neighbor machine learning framework. For an inflected word form, the nearest neighbor algorithm looks up the root form among its "neighbors" by making replacements to its affixes. The Septuagint corpus created by the University of Pennsylvania is used as the training and test data sets. The whole Septuagint, except the first five books, contained 470K words, and 37K unique words were used for the training set. The first five books, which contained 120K words and 3,437 unique words, were used as the test set. To perform the prediction of novel roots, they used the Thesaurus Linguae Graecae corpus, which contained more than one million unique words. The overall accuracy of the proposed system is about 85.7%.

Anand Kumar et al. (2010) [5] presented a new approach to implement Morphological Analyzer for Tamil Language using corpus-based machine learning methodology. Here, a novel method for the morphological analyzer of the Tamil language is implemented based on sequence labeling, and training is done by kernel methods that identify the nonlinear relationships of the morphological characteristics from training data samples in an efficient way. In order to compare the results, Support Vector Machine (SVM), Conditional Random Fields (CRFs), and Memory Based Tagger (MBT)(*k*-NN) algorithms have been applied to the training and testing of the system. In this machine learning approach, morphological analysis is performed based on two training models. These two models are represented as a segmentation model and a morpho-syntactic tagging model. The open-source software packages called TiMBL, CRF++, and SVMTool are used to implement the system. To implement the morphological analyzer for verbs and nouns, the system is trained with 130,000 and 70,000 words, respectively. The system is also tested with 40,000 verbs and 30,000 nouns from an Amrita POS Tagged corpus. The SVM-based machine learning tool obtained better performance compared to CRF and MBT. MBT takes short training time compared to SVM and CRF++. The accuracies of CRF, MBT, and SVM algorithms on the test data are about 89.72%, 90.38%, and 93.65%. MAG for other Dravidian languages like Malayalam, Telugu, and Kannada have been developed using the same methodology.

Canasai Kruengkrai et al. (2006) [6] proposed another statistical model for the Thai language. They considered MA as a search problem that was formulated based on a framework called conditional random fields (CRFs) algorithm. Thai POS-tagged ORCHID corpus, which contained 23125 sentences, is used in this experiment. They split the corpus into 80% for training and the remaining 20% for testing. For a given sequence of characters, all possible word/tag segmentations are generated to select the optimal path with some standards. The evaluation of the ORCHID corpus shows that selecting the optimal path with the confidence estimation is very promising, and results showed 79.744% of precision, 79.744% recall, and 79.342% F-score.

Jedrzejowicz and Strychowski et al. (2005) [9] implemented a morphological analyzer based on a neural

network (ANN) to evaluate the inflection of Polish words. The main tasks of the analyzer are to create base forms of words from the analyzed word forms and to generate grammatical information for the analyzed forms. In this method, every word form is assigned to a valid pattern, which consists of a set of affixes. Every possible inflectional pattern is stored in the base of the inflection patterns. Processing of MA is simply to find a valid inflection pattern for the analyzed form. To optimize searching in the inflection pattern set, they are stored as a decision tree. In order to solve the problem about choosing a valid pattern from all possible candidates returned by the decision tree is to use an artificial neural network as a classifier. In the morphology analysis, the neural network selects the valid inflection pattern from all the candidates returned by a decision tree. The backpropagation algorithm trains neural networks with good results. The presented morphological analyzer was developed as a part of the ICONS project, and it can be extended as stemmer, full-text search engine to search words written in the Polish language. Using a neural network-based analyzer with the extended decision tree and Backpropagation algorithm (ANN), the words are inflected with a very high quality of 99.9 %.

Premjith et al. (2018) [10] proposed a deep learning approach for learning the rules for automatically classifying morphemes and segmenting them from the original word. Then, each morpheme can be further analyzed to classify the grammatical structure inside the word. In the first instance, morphological paradigms for nouns and verbs were developed by categorizing them from the corpus containing 66,601 nouns and 87,676 verbs. Words are then split into characters to acquire 11,06,810 characters in noun corpus and 13,14,198 characters in verbs to implement a character-level morphological analyzer. Three different systems were tested for this analysis based on Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). The system achieved accuracies of 98.08%, 97.88%, and 98.16%, respectively.

Antony et al. (2010) [7] developed a paradigm-based morphological analyzer for a highly agglutinative Kannada language using a machine learning approach. Paradigm-based approaches are appropriate for inflectionally and agglutinatively rich languages. In this approach, the extreme support of a linguist or the language expert is required to provide different tables of word forms that cover the words in a particular language. It is reported that MAG systems have been developing for most of the Indian languages using a paradigm-based approach. This approach has two types of databases which are grouped into sets called paradigms: a lexicon of stems and a separate database of affixes. The paradigms or sets of every word form have grammatical information for all suffixes. There is a separate lexicon for words of the closed word classes as they do not have any paradigm information. Each Paradigm has a set of add-delete rules to evaluate its inflections. A paradigm-based morphological analyzer extracts stem, parts of speech, and inflectional information of word from the inflected word. The implementation of the Kannada morphological analyzer is designed using a sequence labeling approach. Training, testing, and evaluations of the system are performed by kernel method based on support vector machine (SVM) algorithms. The proposed system identifies the nonlinear relationships and the morphological structures of the Kannada language in a very efficient way. The experiment showed that the performance of the system obtained a very high accuracy of 96.25% for morphological analysis of Kannada verbs.

Dikshan and Harshad (2020) [8] developed a Morph analyzer program for the Gujarati language based on a paradigm-based approach. A morph analyzer program for the Gujarati language has been developed to perform a syntactic analysis of a word and to extract the root form of an inflected word. In order to obtain a higher understanding of a language, word level, sentence level, context level, and discourse level, analysis must be done. One of the main purposes is the morphological level analysis for various word forms. A morph analyzer program for the Gujarati language has been developed to perform morphological analysis. The algorithmic development shows an accuracy of 84.50% for nouns, 81.50% for verbs, and 80.50% for adjectives.

Kimmo Koskenniemi, a Finnish computer scientist **(1983) [11][12]** developed a computational linguistics model for word-form identification and generation called two-level morphology using Finite State Transducer (FST). Finite State Transducer (FST) is a modified version of Finite State Automata (FSA). FSA is a device that is composed of states, transitions, and actions. It can accept or reject a string in each language. Finite State Transducer (FST) is two tapes of FSA that can accept input and generate output. The variety of languages can be handled by the two-level morphology using FST. A two-level model using Finite State Transducer (FST) consists of a lexical and a surface representation of a word. The lexical level represents the structure of the functional components of a word, while the surface level represents the actual realization of the word. In two-level morphology, the sequencing of the morphemes and affixes is encoded as finite state machines. It is based on parallel rules and not on formal cascade rules. Lexical look-up and morphological analysis are performed in parallel. It can handle regular expressions as an input string, which makes it a powerful tool for text searching and text recognition. The theory of two-level morphology was shown to be successful for many morphologically rich languages. Subsequently, researchers have taken up a two-level model to develop MAG for many other Asian languages.

Katushemererwe and Hanneforth (2010) [13] presented a computational model for grammatical Runyakitara verbs based on a two-level finite-state method. Morphological analysis of Runyakitara Verbs is implemented using the freely available open-sourced fsm2 interpreter, a scripting

language within the framework of finite-state technology. fsm2 can load lexicons, grammars and replace rules defined by linguists or the morphology developer. It can automatically transform various rule formats into transducers. A list of 3971 words extracted from the dictionary and a Runyakore orthography reference book (Taylor,1985) was used for testing the system. The precision of 82% is proof that the fsm2-based approach is applicable to a morphologically complex Runyakitara, Bantu language.

Sarveswaran et al. (2019) [14] have developed ThamizhiFST, which can analyze more than 800,000 surface forms of Tamil verbs. ThamizhiFST for the Tamil language has been developed using the FOMA toolkit, which supports XFST and can be integrated into Lexical Functional Grammar (LFG) that is written using XLE. The coverage can be improved by adding new verbs to the respective class files. Their implementation of the ThamizhiFST resulted in a precision of 97%. This score is satisfactory for a morphologically rich language like Tamil.

Sarveswaran et al. (2019) [15] developed a Finite-State Morphology (FSM) in the context of computational grammar based on the standards and methods of the international ParGram effort. They introduced a system of meta-morph(ology) rules along with a script that helped to speed up the development of the Tamil FSM. The Tamil FSM tool for Finite-State Morphology was developed using open-source software called FOMA. It can be used with Xerox's XFST compiler and can be integrated to Lexical Functional Grammar (LFG). Tamil FSM can analyze the inflectional morphology of approximately 100,000 nouns and 3300 verb roots with 260-word forms. The Tamil FSM showed a high accuracy for the analysis of verbs and an acceptable accuracy for the analysis of nouns. The accuracies of Verbs and Nouns are nearly 90 %.

Ayla Kayabaş et al. (2019) [16] developed TRMOR, a morphological analyzer for Turkish using the SFST tool (Stuttgart Finite-State Transducer). It covers a large part of Turkish morphology, including inflection, derivation, and compounding. SFST tool needs morphophonological rules and a stem lexicon to implement the system. The evaluation of TRMOR was performed using gold-standard words. For the evaluation process, a corpus with 2 million words extracted from Wikipedia, the Turkish version of Wikipedia, was used. TRMOR reported 94.12% precision on 1000 words that were randomly selected from Wikipedia word lists.

Researchers have implemented Morphological analysis and generation using popular tools based on Finite State Algorithms. **(Beesley and Karttunen, 2003) [17]** presented the use of **Xerox Tools and Techniques.** This toolkit includes three compilers: LEXC, a high-level language for identifying lexicons, TWOLC, a high-level language for identifying morphotactic rules or orthographic rules; and XFST, an interface for building and manipulating regular-expression and finite-state networks. **Cyril Allauzen et al. (2007) [18]** presented an openFST toolkit, a library based on weighted finite-state transducers (FSTs). OpenFST was developed by contributors from Google Research and NYU's Courant Institute. HFST **(Lindén et al., 2009) [19]** and Foma **(Mans Hulden, 2009) [20]** are also efficient tools for creating and manipulating finite-state Morphology for different languages. Wolaytta, an Omotic language of Ethiopia, was developed using the Helsinki Finite-State Transducer toolkit (HFST). Evaluation of the transducer showed high precision (94.85%) and recall (94.11%). **(Abebe et al. 2018) [21]**. **Rahman and Sarma (2015) [22]** have used Apertium lttoolbox to develop a morphological analyzer for the Assamese Language and achieved 72.7% accuracy.

Anna Zueva et al. (2020) [23] created another morphological analyzer for Evenki, a language with rich morphology. The Helsinki Finite-State Transducer toolkit (HFST) is utilized as a primary toolkit to implement a Finite-State Morphological analyzer for Evenki, one of the Russian Languages. The Helsinki Finite-State Transducer toolkit (HFST) supports LEXC formalism to assign the valid orderings of morphemes in words and to determine the morphotactic rules. Morphophonological alternations and orthographic rules are stated using TWOL formalism. Newspaper corpus, Linguistic Corpora at IEA RAS, and Siberian Lang copra are utilized for the development and evaluation of the analyzer. The lexicon collection was carried out using available machine-readable dictionaries. The system achieved coverage scores of between 61% and 87%.

Ammari and Zenkouar., (2021) [24] presented an Amazigh pronominal morphological analyzer (APMorph) using Xerox's finite-state transducer (XFST). Xerox's XFST tool remains relevant for MAG as it allows both analysis and generation. A large lexicon called "APlex," including 12000 nouns,16000 verbs, and the features relating to each lemma, is defined in the XFST notation. This system was able to recognize most categories of affixed pronouns to the verbs and nouns (24440 words of 28000 total words), which proves the success of this approach. An Amazigh pronominal morphological analyzer (APMorph) resulted in an accuracy of 87 %.

Amr Keleg et al. (2020) [25] devised a method for weighting a morphological analyzer using finite state transducers, based on a word2vec as a word embedding model. The system is trained in a completely unsupervised manner using raw untagged corpora and can capture the semantic meaning of the words. The usage of word2vec as a word embedding model to disambiguate the analyses generated by an FST is completely unsupervised and uses information about the word only, irrespective of its context. They have shown that directly weighting the analyses of words instead of relying on the context to disambiguate the results is a successful technique to deploy. Wikipedia dumps for English, Kazakh, and Serbo-Croatian are used to train models. The evaluation process relied on Apertium's tagged corpora for English, Kazakh, and Serbo-Croatian. These

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

corpora are distributed under the GNU General Public License. The word2vec model was very successful in disambiguating English analyses and shows an accuracy of 70.8% for the English language. This method proved to be helpful for both morphological analysis and tagging tasks.

Soe Lai Phyu and Aye Thida (2012) [26] presented the ML2KR framework, which consists of a Morphological processor, Myanmar WordNet, and a bilingual Lexicon. This report focused on inflectional morphology and proposed a twofold rule-based Morphocon: a morphological analyzer for Myanmar words and a morphological generator for translating Myanmar words to English using FSA. Morphocon is tested with 100 sentences with word lengths between 5 and 15. They reported that the result was satisfactory. Precision, recall, and f-measure are nearly 95% in the morphological analyzer and generator. However, this research project is not publicly available to be tested and integrated into other natural language processing (NLP) applications, especially spellchecking, machine translation, and information retrieval.

T. M. Latt and A. Thida (2018) [27] presented the Finite State Automaton (FSA) model for morphological analysis to identify the inflectional morphemes if a word is singular or plural on noun, singular or plural on pronoun, a positive or comparative or superlative form on adjective and present tense or past tense or future tense on verb. FSA is a single-tape model and is not likely to be used for MAG. This FSA-based system could not generate grammatical information for analyzed words. FSA can only perform "checking" a given string to determine whether it can be accepted or rejected. FSA cannot provide any output. Small-scale inflectional morphology was tested using 15024 words, and more than 90 % of words are well-inflected.

Sgarbas et al. (2000) [28] presented a language-independent lexicon-based Directed Acyclic Word Graph (DAW) approach to solve the problem of Greek morphology. DWAG is capable of storing interconnected word forms, lemmas, grammatical information, and word morphological structures. The structure is like a lexical transducer, which has two layers called surface and lexical forms, as well as morphological information. It can store finite strings of information in a compressed way. Thus, MAG can be processed very fast even if many parts of the information cannot be searched from the input. The presented approach is language-independent, and the system does not require any other special linguistic information as well as morphotactic rules. This technique is used in the project MITOS, and the analyzer can perform 10.000 words/sec faster than a Greek two-level morphological analyzer.

Anup Kumar Barman et al. (2019) [29] proposed a Dictionary Look-up and Rule-based suffix-stripping approach for the Assamese language, one of the Indo-Aryan languages. This approach used the dictionary with the root words in the back end. The dictionary, which is about 2 lakh root words, was built by the linguists from Assamese WordNet and named entities. It uses a set of replacement rules to look up the root word from a given collection of words. The suffixes from the end of a word are removed repeatedly until the word is found in some dictionaries. The advantage of using this approach is that it is simple to develop, but its complexity is very high as the word must be searched in the dictionary every time. Thus, a large lexicon dictionary must be built. This method is suitable for some languages which are less inflectional and have a smaller number of suffixes. Stemming or suffix stripping is essential to modern Information Retrieval systems and morphological analysis. This system reports an accuracy of 85%. The analyzer can identify the sentence types and word categories based on the affix information. The better accuracy depends on the size of the dictionaries.

Khumbar Debbarma et al. (2012) [30] presented a design and implementation of a Kokborok morphological analyzer based on a database-driven affix stripping algorithm. Kokborok, one of the Indian languages, has a complex agglutinative structure of words. For every input sentence, the system analyzes each surface-level word to extract the root word and related information, such as the lexical category of the roots and affixes, with the help of morpheme dictionaries. Three modules, Tokenizer, Stemmer, and Morphological Analyzer, are integrated as morphological analyzers. The collection of 56732 Kokborok words has been used to test the Morphological Analyzer, and an accuracy of 80% has been reported.

III. COMPARISON

According to the literature, researchers have taken different techniques, which have advantages and disadvantages for the implementation process.

Table 1: Comparison of Different Morphological Techniques

Comparison of Different Morphological Techniques					
Techniques	Method	Advantages	Disadvantages	Related Work	Result
Corpus-Based Machine Learning Approach	- Backpropagation algorithm (ANN) - Support Vector Machine (SVM) algorithm - Conditional Random Fields (CRF) algorithm	- corpus can be used in the Statistical model for supervised or unsupervised machine learning approaches. - results are good if sufficient data is available	- need a large amount of data for machine learning purposes - results depend on the content of the corpus used.	(i) Premjith et al., (2018) [10] (ii) Jedrzejowicz and Strychowski et al., (2005) [9]	98.08% 97.88% 98.16% 99.9 %

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

Comparison of Different Morphological Techniques					
Techniques	Method	Advantages	Disadvantages	Related Work	Result
	- Memory-Based Tagger (MBT)(*k*-NN) algorithms - Nearest Neighbor machine learning framework - Minimum Description Length algorithm (MDL) - Recurrent Neural Network (RNN) - Long Short-Term Memory (LSTM) - Gated Recurrent Units (GRU)	- produces better results if a language has special corpora.	- making a corpus is time-consuming.	(iii) Antony et al., (2010) [7] (iv) Canasai Kruengkrai et al., (2006) [6] (v) Anand Kumar et al., (2010) [5] (vi) John Lee (2008) [4] (vii) Viraj Welgama et al., (2013) [3] (viii) John Goldsmith (2005) [2]	96.25% 79.7% 89.72%, 90.38% 93.65% 85.7% 56% 72%
Finite State Machine Approach	- Finite State Automata - Finite-State Transducers - Two-level morphology - weighted FST with word2vec model	- good for lexicon recognition and computational linguistics. - could perform MAG. - several tools are available for development. - Only lexicon and rules are needed to perform MAG. - It is a rule-based technique. - MAG can be performed by various tools like XFST, HFST ,FOMA, openFST, Apertium	- FSA is not good for MAG - FSA can't give any output. - set of rules will lead to ambiguity. - if rules followed by previous rules fail, a series of failures may occur.	(i) T. M. Latt and A. Thida (2018) [26] (ii) Amr Keleg et al., (2020) [25] (iii) Ammari and Zenkouar., (2021) [24] (iv) Anna Zueva et al., (2020) [23] (v) Rahman and Sarma., (2015) [22] (vi) Abebe et at., (2018) [21] (vii) Sarveswaran et al., (2019) [15] (viii) Sarveswaran et al., (2019) [14] (ix) Katushemererwe and Hanneforth., (2010) [13]	95% 70.8% 87 % 87 % 72.7% 94.85% 90 % and above 97% 82%
Paradigm-based Approach	- paradigm tables - Support Vector Method (SVM) algorithms	- suitable for agglutinative language - improved results with well-defined paradigm table	- the result is based on the content of the paradigm table. - the same word may possess many features - need expert or linguists	(i) Antony., (2010) [7] (ii) Dikshan and Harshad., (2020) [8]	96.25% above 80%
Directed Acyclic Word Graph	- the finite strings are stored as directed paths on the graphs.	- language-independent and do not use any morphological rules	- can move only in the direction specified	(i) Sgarbas et al., (2000) [28]	~

Comparison of Different Morphological Techniques					
Techniques	Method	Advantages	Disadvantages	Related Work	Result
Suffix Stripping	- the suffixes from the end of the word are removed recursively until the word is found in some dictionaries	- simpler to maintain and is very fast to remove double words.	- cannot provide the relation of words that have different word forms according to their grammatical structure.	(i) Anup Kumar Barman et al., (2019) [29] (ii) Khumbar Debbarma et al., (2012) [30]	85% 80%

IV. CONCLUSION

The morphological analysis and generation (MAG) task is the primary component of computational linguistics and language processing. According to the literature, researchers have utilized different approaches to implement different models in developing morphological analysis and generation. We can conclude that the most appropriate method is a finite-state method, which has been shown to be more relevant to languages with poor linguistic information. Finite State Morphology is based on three main components: morpheme lexicon, morphotactic or orthographic rules, and decision algorithms.

Myanmar language is a highly agglutinative language and linguistically complex. As for the Myanmar language, there is a lack of research reports and developments available for morphological analysis and generation. Myanmar language still doesn't have any morphological analyzer or generator that is publicly available. More research is still needed to implement a morphological analyzer and generator to be used for other NLP tasks. As further work for this paper, a full-fledged morphological analyzer and generator that can cover in-depth linguistic phenomena will be developed based on finite-state methods.

REFERENCES

[1] Goldsmith, John. (2001). Unsupervised learning of the morphology of a natural language. Computational linguistics, 27(2):153–198, 2001.

[2] Goldsmith, John. (2006). An Algorithm for the Unsupervised Learning of Morphology. Natural Language Engineering. 12. 353-371. 10.1017/S1351324905004055.

[3] Welgama, Viraj & Weerasinghe, Ruvan & Niranjan, Mahesan. (2013). Evaluating a Machine Learning Approach to Sinhala Morphological Analysis. 10th International Conference on Natural Language Processing, At: Noida, India

[4] John Lee. (2008). A Nearest-Neighbour Approach to the Automatic Analysis of Ancient Greek Morphology. CoNLL 2008: Proceedings of the 12th Conference on Computational Natural Language Learning, Manchester, August 2008: 127–134

[5] Anand Kumar, M. & Dhanalakshmi, V. & Kp, Soman & Sankaravelayuthan, Rajendran. (2010). A Sequence Labeling Approach to Morphological Analyzer for Tamil Language. (IJCSE) International Journal on Computer Science and Engineering Volume 02, Issue No. 06, Page no (2201-2208),2010.

[6] Canasai Kruengkrai, Virach Sornlertlamvanich, Hitoshi Isahara. (2006). A Conditional Random Field Framework for Thai Morphological Analysis. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-06), May 24-26, 2006. Genoa, Italy.)

[7] P. J. Antony, Dr. M. Anand Kumar, and Dr. Soman K. P. (2006). Paradigm based morphological analyzer for Kannada language using machine learning approach. International journal on-Advances in Computer Science and Technology (ACST), ISSN 0973-6107, vol. 3, pp. 457–481.

[8] Shah D.N., Bhadka H. (2020). Paradigm-Based Morphological Analyzer for the Gujarati Language. Advances in Intelligent Systems and Computing, vol 989. Springer, Singapore. https://doi.org/10.1007/978-981-13-8618-3_50

[9] Jedrzejowicz, Piotr & Strychowski, Jakub. (2005). A Neural Network Based Morphological Analyser of the Natural Language. 199-208. 10.1007/3-540-32392-9_21.

[10] 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.

[11] Kimmo Koskenniemi. (1983). Two-level morphology. Ph.D. thesis, University of Helsinki.

[12] Lauri Karttunen and Kenneth R Beesley. (2012). A short history of two-level morphology. ESSLLI-2001 Special Event titled" Twenty Years of Finite-State Morphology. Available: http://www.helsinki.fi/esslli/.

[13] Katushemererwe, F., & Hanneforth, T. (2010). Finite State Methods in Morphological Analysis of Runyakitara Verbs. Nordic Journal of African Studies, 19, 22-22.

[14] Sarveswaran, K., Dias, G., & Butt, M. (2018). ThamizhiFST: A Morphological Analyser and Generator for Tamil Verbs. 2018 3rd International Conference on Information Technology Research (ICITR), 1-6.

[15] Kengatharaiyer, Sarveswaran & Dias, Gihan & Butt, Miriam. (2019). Using Meta-Morph Rules to develop Morphological Analysers: A case study concerning Tamil. 10.18653/v1/W19-3111.

[16] Kayabaş, A., Schmid, H., Topcu, A., & Kiliç, Ö. (2019). TRMOR: a finite-state-based morphological analyzer for Turkish. Turkish Journal of Electrical Engineering and Computer Sciences, 27, 3837-3851.

[17] Kenneth R Beesley and Lauri Karttunen. (2003). Finite-state morphology: Xerox tools and techniques. CSLI Publications, Stanford.

[18] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. (2007). OpenFst: A general and efficient weighted finite state transducer library. In International Conference on Implementation and Application of Automata, pages 11–23. Springer.

[19] Krister Lindén, Miikka Silfverberg, and Tommi Pirinen. (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, pages 28–47. Springer.

[20] Mans Hulden. (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, pages 29–32. Association for Computational Linguistics.

[21] Abebe, Tewodros & Washington, Jonathan & Gasser, Michael & Yimam, Baye. (2018). A Finite-State Morphological Analyzer for Wolaytta. Information and Communication Technology for Development for Africa.10.1007/978-3-319-95153-9_2.

[22] Rahman, Mirzanur & Sarma, Shikhar. (2015). An Implementation of Apertium Based Assamese Morphological Analyzer. International Journal on Natural Language Computing. 4. 10.5121/ijnlc.2015.4102.

[23] Zueva, A., Kuznetsova, A., & Tyers, F. (2020). A Finite-State Morphological Analyser for Evenki. Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2581–2589.

[24] Ammari, Rachid & Zenkoua, Ahbib. (2021). APMorph: finite-state transducer for Amazigh pronominal morphology. International Journal of Electrical and Computer Engineering (IJECE). 11. 699. 10.11591/ijece.v11i1.pp699-706.

[25] Keleg, A., Tyers, F.M., Howell, N., & Pirinen, T.A. (2020). An Unsupervised Method for Weighting Finite-state Morphological Analyzers. Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 3842–3850

[26] Phyue, S. L., & Thida, A. (2012). Morphological Processor for Inflectional Case of Multipurpose Lexico-Conceptual KnowledgeResource. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1(7), 157-163.

[27] Latt, T. M., & Thida, A. (2018, June). An Analysis of Myanmar Inflectional Morphology Using Finite-state Method. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) (pp. 297-302). IEEE.

[28] Sgarbas. (2000). A Straight Forward Approach to Morphological Analysis and Synthesis, Sgarbas (et al. l), Proceeding of COMLEX 2000, Greece.

[29] Sarmah, J., Sarma, S.K., & Barman, A. (2019). Development of Assamese Rule-based Stemmer using WordNet. GWC.

[30] Debbarma, K., Patra, B. G., Das, D., & Bandyopadhyay, S. (2012, December). Morphological Analyzer for Kokborok. In Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (pp. 41-52).