**Morphological Analyzers: A Review**

Shubhnandan Singh Jamwal

Department of Computer Science and IT, University of Jammu, J&K

**Abstract:** The development of Morphological Analyzer (MA) for the Indian language is a challenging task and it involves the detailed analysis of taking a word as input by a model and identifying their stems and affixes. The Morphological analysis of a language provides information about words, semantics and its syntactic role in a sentence. In the development of Morphological Analyzer morphemes in the language, the rules how these morphemes are connected and the changes when prefixes and suffixes are added to a morpheme are very important considerations. In this paper, a review of the different languages especially Indian, a review is presented for the development of the MA.

**Keywords:** Morphological Analyzer, Tokenization, Normalization, Word Segmentation, POS tagging, Indian languages.

**Introduction**

India is a rich country of different cultures where numbers of different languages are spoken and Indian government is trying to manually translate among 22 official languages, but assistance, in the form of advanced technology enabled by a lot of hard work, is on the way [1]. The development of complete natural language processing (NLP) tools and models for any language with limited resources is always a challenging and demanding task. Hindi is an official language of India and number of researchers in the India trying to develop the morphological analyzers. Hindi is relatively rich in morphology and few morphological analyzers of this language have been developed. The MA of Hindi gives inflectional analysis of official language.

The Morphological Analyzer and Generator is an indispensable tool for the digitalization of any language and these tools are also extensively required in developing NLP applications such as spell checker, Machine Translation (MT), Information Retrieval (IR), etc. A MA describes the root word and its various forms morphemes ordering structure of the word form in use. K. V. N. Sunitha and A. Sharada [5] observed that India is very rich in heritage and culture. There are many ancient scientific and invaluable literatures which reflect our age old culture and prove India is scientifically very much advanced than any other country hundreds and thousands of centuries back itself. In order to preserve this knowledge and make it available to all the people there is need to digitize this literature. In this paper we discuss the problem of this issue and propose a model which can be used to overcome the problem. Main claim of the paper is proposal of architecture for implementing the solution. The morph length is one of the indicative feature that helps in learning the morphology of languages and this feature is exploited by number of researchers in the development of NLP tools.

**Literature Review**

Nikhil Kanuparthi, Abhilash Inumella, and Dipti Misra Sharma [2] presented Hindi derivational morphological analyzer. Their proposed algorithm upgrades an existing inflectional analyzer to a derivational analyzer and primarily achieves two goals. First, it successfully incorporates derivational analysis in the inflectional analyzer. Second, it also increases the coverage of the inflectional analysis of the existing inflectional analyzer.

P. J. Antony, Hemant B. Raj, B. S. Sahana, Dimple Sonal Alvares, and Aishwarya Raj[3] presented the development of a Morphological Analyzer and Generator for an agglutinative Dravidian language called Tulu. Even though such a Morphological Analyzer and Generator tool is mandatory for many Natural Language Processing (NLP) tasks but is not yet available for Tulu language. In the proposed rule based Morphological Analyzer and Generator system is implemented using AT & T Open FST (Finite state transducer). From the experiment it is observed that, the performance of the proposed system is encouraging and can be substantially improved by increasing the number of rules.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari [4] while doing research of Arabic languages designed tools and resources for natural language processing of Arabic are designed for Modern Standard Arabic (MSA) and perform terribly on Arabic dialects, such as Egyptian Arabic. Egyptian Arabic differs from MSA phonologically, morphologically and lexically and has no standardized orthography. We present a linguistically accurate, large-scale morphological analyzer for Egyptian Arabic. The analyzer extends an existing resource, the Egyptian Colloquial Arabic Lexicon, and follows the part-of-speech guidelines used by the Linguistic Data Consortium for Egyptian Arabic. It accepts multiple orthographic variants and normalizes them to a conventional orthography.

Loganathan Ramasamy, Zdeněk Žabokrtský, and Sowmya Vajjala [6] introduced a simple unsupervised model for morphological segmentation and studied how the knowledge of morph length affects the performance of the segmentation task under the Bayesian framework. The proposed model is based on (Goldwater et al., 2006) unigram word segmentation model and assumes a simple prior distribution over morph length. They performed the experiment of this model on two highly related and agglutinative languages namely Tamil and Telugu, and compare our results with the state of the art Morfessor system and observed that, knowledge of morph length has a positive impact and provides competitive results in terms of overall performance.

Akshar Bharati and Rajeev Sangal [7] described a karaka based approach for parsing of Indian languages which has been used for building a parser of Hindi for a prototype Machine Translation system. A lexicalised grammar formalism has been developed that allows constraints to be specified between 'demand' and 'source' words (e.g., between verb and its karaka roles). The parser has two important novel features: (i) It has a local word grouping phase in which word groups are formed using 'local' information only. They are formed based on finite state machine specifications thus resulting in a fast grouper. (ii) The parser is a general constraint solver. It first transforms the constraints to an integer programming problem and then solves it.

Peter McClanahan, George Busby, Robbie Haertel, Kristian Heal, Deryle Lonsdale, Kevin Seppi, and Eric Ringger [8] defined a probabilistic morphological analyzer using a data-driven approach for Syriac in order to facilitate the creation of an annotated corpus. Syriac is an under-resourced Semitic language for which there are no available language tools such as morphological analyzers. The researcher introduced probabilistic models for segmentation, dictionary linkage, and morphological tagging and connect them in a pipeline to create a probabilistic morphological analyzer requiring only labeled data. They also explored the performance of models with varying amounts of training data and find that with about 34,500 labeled tokens, we can outperform a reasonable baseline trained on over 99,000 tokens and achieve an accuracy of just over 80%. When trained on all available training data, our joint model achieves 86.47% accuracy, a 29.7% reduction in error rate over the baseline. V. Goyal and G. S. Lehal [9] presented the morphological analysis and generator tool for Hindi language using paradigm approach for Windows platform having GUI. This project has been developed as part of the development of a machine translation system from Hindi to Punjabi Language.

P. Das and A. Das [10] described a linguistics approach towards development of a Bengali Noun Morphological Analyzer implemented at first on the semi-manually created database of 87697 inflected words list tokens, i.e. Input2 for Linguistics Resource Creation comprising of Noun, Pronoun, Adjective roots with and without its suffixes. Then after the first implementation the developed Linguistic Resource knowledge is applied on an unknown Bengali corpus database containing 6157 tokens. At the initial stage of this research a linguistic analysis is done which leads to framing of the nominal suffix list which is later on used in nominal suffix extraction. This linguistic knowledge is implemented in developing the finite-state transducer grammar for Linguistic Resource which gives way to the development of Bengali Noun Morphological Analyzer. The final output obtained is around 44% accuracy. This accuracy can be always improved with time if we keep on increasing the nominal roots in the FST grammar file.

S. Lushanthan, A. R. Weerasinghe and D. L. Herath [11] developed a “Morphological Analyzer and Generator for Tamil Language” that will be generating the word forms of a stem/ root, given a particular context and at the same time, a surface form in Tamil language should get analyzed into its proper context. This model tries to cover only the nouns and verbs in the Tamil language. They illustrated that how the lexicon and the orthographic rules of Tamil language have been written as regular expressions using only finite state operations and how this approach has been implemented to develop a morphological analyzer/generator. The proposed model is built using the Xerox toolkit, which uses “Two-level Morphology”, and almost 2000 noun stems and 96 verb stems have been incorporated into the network. A noun stem now produces about 40 different forms and a verb stem produces up to 240 forms. We have also defined our own transliteration scheme for this purpose.

**Phases of Morphological Analyzer**

Morphological analysis is an essential component in Natural Language Processing (NLP) applications ranging from spell checker to machine translation. The design and development of the morphological analyzer consists of four phases. These are word tokenization, normalization, word segmentation, and POS tagging. The first phase of a MA is a tokenization in the development of any NLP tools. This phase is a text processing process that divides paragraphs into sentences and sentences into words and while performing a morphological analysis it also leads to segmentation of a word into morphemes. The second phase is the normalization process that consists of several text processing algorithms, such as decoding abbreviations, clearing text from punctuation marks, converting characters to a single case etc. The segmentation is the third phase of the MA which involves the division of words into root, prefix, suffix, affixes etc. The next phase in the development of the MA is POS tagging. The phase uses the technique of employing the POS tag to each word. POS tagging is the process of determining the part of speech and grammatical properties of words and on the basis of this POS tag is employed.

**Conclusion**

Morphology of languages is the field of the linguistics that studies the internal structure of the words. The internal structure of a word can also be used for the many NLP application development tools. Morphological analysis and generation are essential steps in any NLP application.

The Indian languages needs Morphological analyzers for the digitalization of the languages. Morphological analyzer and generator is a tool for analyzing the given word and generator for generating word given the stem and its features. In this paper, number of techniques which are used by the researchers for the development of the MA are discussed. The performance of the MA also varies from language to language. The development of MA of highly inflectional language needs a very deep understanding of the language by the researcher.

**References**

[1] Gary Anthes. 2010. Automated translation of Indian languages. Commun. ACM 53, 1 (January 2010), 24–26. <https://doi.org/10.1145/1629175.1629184>

[2] Nikhil Kanuparthi, Abhilash Inumella, and Dipti Misra Sharma. 2012. Hindi derivational morphological analyzer. In Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON '12). Association for Computational Linguistics, USA, 10–16.

[3] P. J. Antony, Hemant B. Raj, B. S. Sahana, Dimple Sonal Alvares, and Aishwarya Raj. 2012. Morphological analyzer and generator for Tulu language: a novel approach. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '12). Association for Computing Machinery, New York, NY, USA, 828–834. https://doi.org/10.1145/2345396.2345531

[4] Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A morphological analyzer for Egyptian Arabic. In Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON '12). Association for Computational Linguistics, USA, 1–9.

[5] K. V. N. Sunitha and A. Sharada. 2010. Digitization of Indian literature: problem and solution. In Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India (A2CWiC '10). Association for Computing Machinery, New York, NY, USA, Article 22, 1–7. <https://doi.org/10.1145/1858378.1858400>

[6] Loganathan Ramasamy, Zdeněk Žabokrtský, and Sowmya Vajjala. 2012. The study of effect of length in morphological segmentation of agglutinative languages. In Proceedings of the First Workshop on Multilingual Modeling (MM '12). Association for Computational Linguistics, USA, 18–24.

[7] Akshar Bharati and Rajeev Sangal. 1990. A karaka based approach to parsing of Indian languages. In Proceedings of the 13th conference on Computational linguistics - Volume 3 (COLING '90). Association for Computational Linguistics, USA, 25–29. <https://doi.org/10.3115/991146.991151>

[8] Peter McClanahan, George Busby, Robbie Haertel, Kristian Heal, Deryle Lonsdale, Kevin Seppi, and Eric Ringger. 2010. A probabilistic morphological analyzer for Syriac. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10). Association for Computational Linguistics, USA, 810–820.

[9] V. Goyal and G. S. Lehal, "Hindi Morphological Analyzer and Generator," 2008 First International Conference on Emerging Trends in Engineering and Technology, Nagpur, India, 2008, pp. 1156-1159, doi: 10.1109/ICETET.2008.11.

[10] P. Das and A. Das, "Bengali Noun Morphological Analyzer," 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 2013, pp. 1538-1543, doi: 10.1109/ICACCI.2013.6637408.

[11] S. Lushanthan, A. R. Weerasinghe and D. L. Herath, "Morphological analyzer and generator for Tamil Language," 2014 14th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 2014, pp. 190-196, doi: 10.1109/ICTER.2014.7083900.