Chapter 2 : REVIEW OF LITERATURE

Morphological analysis can be done in many ways as is available in the literature. Corpus-based, paradigm-based, rule-based, finite-state automata based, two-level morphology, finite-state transducer-based approaches are some of the popular methods reported in the literature.

Corpus-based approaches require well developed annotated language corpus. Languages that have such resources can be processd using statistical methods. Statistical approaches to morpheme segmentation depend on the training of hypothetical models, which requires excessive amounts of data, from few hundred thousand to millions of words. Morphological analyzer and generator have been developed for major European languages with well populated text corpora. In these methods the training data are labeled, unlabeled, or partially labeled respectively. And it is obvious that the success rate depends on the size and coverage of the corpus in question.

John Goldsmith (2001) proposed Linguistica, a corpus-based method for the learning of the morphology of some European languages with a goal to treat unrestricted natural languages. Corpora ranging in size from 5,000 words to 500,000 words were used for the purpose. It assumes suffix-based morphology. A set of heuristics that rapidly develop a probabilistic morphological grammar and minimum description length (MDL) was the primary tool to determine whether the modifications proposed by the heuristics will be adopted or not. It performs unsupervised learning in the sense that the program's sole input is the corpus; the program is provided with the tools to analyze with no dictionary and no morphological rules. It is claimed that the resulting grammar matches well with the analysis that would be developed by a human morphologist. Morphological analyzer for English, French, Italian, Spanish, and Latin have been developed using this approach.

Two methods namely- Minimum Description Length (MDL) and maximum likelihood optimization are used for unsupervised segmentation of morphemes by Mathias Creutz, et al. (2002)**.** Here Goldsmith's Linguistica and the two methods were tested on Finnish and English corpora for a comparison of performance. In case of Finnish, the recursive MDL method has better data compression producing smallest morph lexicon, occupying a small part of the total cost; Goldsmith's Linguistica, on the other hand, employs a more restricted segmentation leading to a larger lexicon and thereby occupying larger part of the MDL cost. However Linguistica is reported to have a faster speed of the two. For English, recursive MDL and Linguistica achieves nearly the same result. The percentage of unseen morph/morphemic label pairs is about the same for all three methods, suggesting that morphologically poor language like English, a restrictive segmentation method such as Linguistica can compensate for new word forms. Though the method by Mathias et. al. (2002), is able to generalize better to new word forms but has somewhat lower accuracy for already observed word forms.

Another statistical model proposed by Canasai Kruengkrai and Hitoshi Isahara (2008), considered morphological analysis as a search problem, that jointly tackles word segmentation, POS tagging, and unknown word problems. Assumption is that the system could analyze a given string accurately (both known and unknown words), if the search space contains reasonably potential word hypotheses. Such space can be generated by using the so-called *two-pass search algorithm.* A dictionary and writing rules to build an initial lattice for a string input is used in the first pass. The suspicious word hypotheses in the lattice are identified and expand new word hypotheses from all possible substrings within uncertainty ranges. In the second pass, the search is for the optimal path in the expanded search space by applying a lattice-based Viterbi algorithm. The two-pass search algorithm is said to have an improvement of over the performance of the standard search by 3.23 *F*1 in word segmentation and 2.92 *F*1 in the combination of word segmentation and POS tagging.

Dang, Minh Thang, and Saad Choudri's (2006) SUMAA, is a hybrid algorithm based on letter successor varieties for an entirely unsupervised morphological analysis.

Isolated and agglutinative languages can be handled using language pattern and structural recognition. However, the results are not so encouraging.

Paradigm based approaches are seen to be applied to inflected languages. Mugdha Bapat et al. (2010), proposes a method for morphological analysis for Marathi. Morphotactics of the language is handled by finite state machines and a system of paradigms to handle the stem alternations. Accuracy is measured manually by counting the number of correctly analyzed words out of the total number of words. A rich lexicon with roots is required for bigger coverage. The result of the system is reported to be well above 90% with and major reasons for recognition failure being root coverage, absence of rules, compound words, and acronyms. Vishal Goyal, et al. (2008), propose another morphological analysis and generator tool for Hindi language using paradigm approach for Windows platform having GUI, as part of the development of a machine translation system from Hindi to Punjabi language. A linguist/ language expert provides different tables of word forms covering the words in the language. A word forms table covers a set of roots and the root follow the pattern (or paradigm) implicit in the table for generating their word forms. The system stores all the commonly used word forms for all Hindi root words in its database, and it is recommended that for the languages in which the number of possible inflections for a word is not infinite or very high. It is also seen in the literature that analysis for agglutinative language like Kannada is also performed using paradigm based approach. A paradigm based morphological analyzer using the machine learning approach is reported also for agglutinative Kannada language by (Antony, P. J., 2010). The analyzer is designed using sequence labeling approach and training; testing and evaluations are done by support vector method (SVM) algorithms. The system is said to have performed competitively well with other openly available systems and a good accuracy of above 95% for Kannada verbs.

Morphological analyzer of many of the South Indian languages of Dravidian family, like Kannada, Malayalam, Tamil, are found in the literature. Ramaswamy, et al.(2011) proposes a rule based morphological analysis with Finite-State Transducers for Kannada. Rule based methods are commonly adopted approaches for these agglutinative languages. However exceptions are also reported in the literature. Jisha

P.Jayan et al. (2011) use a bilingual dictionary for Malayalam and Tamil which consist of the root/ stem of the words with its grammatical category for the proposed system. The Malayalam morphological analyzer and the Tamil morphological generator were developed for the purpose of Malayalam - Tamil Machine Translation. For the analysis, suffix stripping method (Malayalam does not have prefix and only suffix) is used to strip off the suffixes attached to the stem of the word and apply proper sandhi rules. For generation of Tamil words, rule definitions are used to generate the desired word form as used for analysis but in the reverse direction. Morphological analysis for Tamil (Dhanalakshmi, et al., 2009) adopted machine learning approach based on sequence labeling and training by kernel methods captures the non-linear relationships in the different aspect of morphological features of natural languages in a better and simpler way.

One of the popularly found approaches for morphological analysis for agglutinative languages is the two-level model of morphology after Kimmo Koskenniemi's doctoral thesis (1983). Since then morphological analyzers have been developed for many languages using this formalism. The advent of two-level morphology Koskenniemi (1983), Karttunen (1983), Antworth(1990), Ritchie *et al.* (1991) has made it relatively easy to develop adequate morphological (or at least morphographical) descriptions for natural languages, clearly superior to earlier "cut-and-paste" approaches to morphology. Harald Trost's (1991) X2MORF is a language independent morphological component for the recognition and generation of word forms based on a lexicon of morphs based on two-level morphology. Word formation is described in a feature-based unification grammar, instead of continuation class. Two-level rules are provided with a morphological context in the form of feature structures. Implementation is done by compiling rules into automata (as in the standard model) and processing of the feature-based grammar enhanced using an automaton derived from that grammar as a filter. The system reportedly runs on CommonLisp on Mac II fx and is used to describe German inflectional and derivational morphology; integrated with a lexicon structure containing lexeme-specific syntactic and semantic information. The variety of languages that can be handled by two-level model using Finite-State Transducers in the literature is a noteworthy one. Celtic language Modern Irish has been treated with this approach

(Dhonnchadha, Elaine Uí, 2002). The morphotactics of stems and affixes are encoded in the lexicon and word mutations are implemented as a series of replace rules encoded as regular expressions and are compiled into finite-state transducers and combined to produce a single lexical transducer for the language. It employs xfst tool for the purpose and the result shows a tremendous performance for modern Irish words for both analysis and generation. Agglutinative Turkish (Oflazer, Kemal, 1994) language adopted the two-level formalism using PC-Kimmo environment, while Malagasy (Mary Dalrymple, et al. 2005) and Arabic (Kenneth R. Beesley, 1996, 2001) uses Xerox tools lexc and xfst for the analysis and generation with the same formalism. JKimmo (Md. Zahurul Islam and Mumit Khan, 2006), is a multilingual morphological open-source framework uses the PC-Kimmo two-level morphological processor and provides a localized interface for Bangla (Bengali) morphological analysis. Finite-State Transducers based morphological analyzers are also reported for the highly inflectional language, Hindi (Deepak Kumar, et al. 2012). A lexicon of root words and rules for generating inflectional and derivational words from these root words, SFST (Stuttgart Finite State Transducer) tool was used for generating the FST. The Morph Analyzer developed was used in a Part Of Speech (POS) Tagger based on Stanford POS Tagger. The system reportedly gave good result.

As for Manipuri language, there are reports of morphological analyzers available in the literature. Th. Doren Singh and Sivaji Bandyopadhyay (2008) uses a Manipuri – English dictionary that stores the Manipuri root words and their associated information, and an affix dictionary which stores the different affixes with their type and the English equivalent pattern of the affix. The work is based on string pattern stripping and matching technique and emphasizes on word compounding, multiple suffix and sentence level dealing. There is repeated access in the suffix table and the overhead is more as the stripping of the morphemes requires testing of various morphemes pattern combinations. Another work on Manipuri is also available. Sirajul Islam Choudhury, et al., (2004) proposes a model to treat orthographic variations, sequential and non-sequential morphotactic constraints and combination of morphosyntactic features. The model is based on the grammatical rules and the root and affix dictionaries. The lexical category of the root and the grammatical category of the affixes are tagged by a model tagger. Morpheme segmentation by stemming, checking the morphosyntactic feature and tagging comprises the main implementation

modules for the purpose. A comparison of the results generated by the morphological analyzer with results generated by human experts made out of their language intuition is claimed to give an accuracy rate of 75%. Implemented using Perl, the working of the system and the outputs claimed to obtain is not so appealing. As of now, no morphological analyzers for Manipuri are reported using the finite-state techniques which are very popular in the literature and known to be suitable for agglutinative languages with complex morphological structures such as Turkish, Manipuri, Basque, Finnish, etc.

2.1 METHODOLOGY

An in depth study finds finite-state automata theory very fascinating for its mathematical properties for a probable application to language processing software. This assumption is supported by an extensive review of the available works on morphological analysis which reveals that the finite-state techniques are the most suitable methodology to perform the task of morphological analysis. This is true for languages with agglutinative word structure with complex word formation processes. We employ finite-state transducer which is a specialized finite-state automaton to describe the morphological analyzer of Manipuri language by fully utilizing the characteristic properties of finite-state techniques. The details of finite-state transducers are discussed in chapter 4.

Our methodology mainly consists to

- identify and list the type of morphemes and their lexical category available in the language.

- define morphotactics of each morpheme (lexical category)

- specification of morphotactics of different word classes in respective source lexicon

- identify spelling change rules for morphophonemic alternations.

- collection of root/stem for language model

For information about morphemes we referred to “Manipuri Grammar” by Ch. Yashwanta Singh and “Manipuri Grammar” by DNS Bhatt and MS Ningomba. Being

a native speaker of the language, my intuition has been used for cross checking and analyzing the data.

The implementation of the language model for computational morphological analysis of Manipuri is done by using Xerox finite-state tools- lexc and xfst (Kenneth R Beesley & Lauri Kartunnen, 2003). A group of lexica (plural of lexicon) has been created following the concept of Two-Level Morphology" defining the morphotactics of the word structure. lexc compiles them to finite-state transducer networks for each word groups and later on into a single lexicon network representing the morphotactics of the word category. Each word category and its own set of orthographic rules are compiled together to a single lexical transducer by composition operation. All the lexical transducers of the word categories are composed together to form one single lexical transducer for morphological processing of all the word classes of the language. See chapter 4 for the detailed description of the theoretical framework.

2.2 DESIGN ISSUES

The study has dealt mainly with the written forms of words in Manipuri. Even then the language does not have a standardized documentation on its spelling. It is crucial to say that the spelling change rules are implemented in such a manner that less modifications is done in case of a change later on. Currently Manipuri is written using Bengali script; so we implemented the system using the same script. However, there is a chance in the near future that Manipuri's own script Meetei Mayek will be used to write Manipuri. This provision is not looked upon for the time being, but it would not be an issue as only the root words need to be converted to Meetei Mayek script from Bengali script for the analysis purpose. All the other symbols such as lexical and grammatical tags are written using Roman script.

2.3 LIMITATIONS

One of the major word formation processes of Manipuri language is compounding. In this study only some compound word forms such as noun + noun and noun + verb forms of compounding cases are taken care of. Also a morphologically category called reduplication is not dealt with in this study. Only the representative words have been considered in the implementation of the analysis.