Automated Paradigm Selection for FSA based Konkani Verb Morphological Analyzer

Shilpa Desai Jyoti Pawar Pushpak Bhattacharya
(1) Department of Computer Science and Technology, Goa University, Goa, India
(2) Department of Computer Science and Technology, Goa University, Goa, India
(3) Department of Computer Science and Engineering, IIT, Powai, Mumbai, India
sndesai@gmail.com, jyotidpawar@gmail.com, pb@cse.iitb.ac.in

ABSTRACT

A Morphological Analyzer is a crucial tool for any language. In popular tools used to build morphological analyzers like XFST, HFST and Apertium's lttoolbox, the finite state approach is used to sequence input characters. We have used the finite state approach to sequence morphemes instead of characters. In this paper we present the architecture and implementation details of a Corpus assisted FSA approach for building a Verb Morphological Analyzer. Our main contribution in this paper is the paradigm definition methodology used for the verbs in a morphologically rich Indian Language Konkani. The mapping of citation form of the verbs to paradigms was carried out using an untagged corpus for Konkani. Besides a reduction in human effort required an F-Score of 0.95 was obtained when the mapping was tested on a tagged corpus.

KEYWORDS: Verb Morphological Analyzer, Corpus, Hybrid Approach, Finite State Automata, Paradigm Mapping.

1 Introduction

Morphological Analysis is a significant step in most Natural Language Processing (NLP) Applications. A Morphological Analyzer (MA) is a tool which retrieves the component morphemes of a given word and analyzes them. Some well known approaches used to build a Morphological Analyzer are Rule Based Affix Stripping Approach, Pure Unsupervised learning of Morphology (Hammarström, 2011), Finite State Approach (Beesley, 2003) and Semi-Supervised learning of Morphology (Lindén, 2009).

Rule based approaches use a set of rules in the language for stemming (Porter, 2000). The success of such approaches will depend on the rules incorporated which are vast for morphologically rich languages. In Pure Unsupervised learning of morphology (Freitag, 2005), (Goldsmith, 2001), (Hammarström, 2011), (Xanthos, 2007) corpus text of the concerned language is used to learn morphology of the language. The reported accuracy of such systems is relatively less as compared to the other methods.

Finite State Transducers is a computationally efficient, inherently bidirectional approach and can also be used for word generation. Many Indian Language groups use finite state tools for morphological analysis. Lttoolbox (Kulkarni, 2010) developed under Apertium is one such tool. The analysis process is done by splitting a word (e.g. cats) into its lemma 'cat' and the grammatical information <n><pl>. The major limitation of such a tool is that it requires a linguist to enter a large number of entries to create a morphological dictionary. It assumes that some form of *word and paradigm* (Hockett, 1954) model is available for the language. This approach to build Morphological Analyzer is time consuming when every word is manually listed in a morphological dictionary and mapped to a paradigm. A funded project to build a Morphological Analyzer using FST tools typically allocate 9 to 12 months for the work. In such Projects mapping of words to paradigms is done by a language expert manually.

In absence of a funded project, limited volunteer human experts are available to manually map words to paradigms. Hence we have made use of existing resources for the Konkani language namely Corpus and WordNet to reduce the human effort to map words to paradigms. For this we have designed our own paradigm structure and FSA based sequencing of morphemes which is used to generate word forms. Our approach builds a morphological dictionary which can be later exported to popular tools like XFST or lttoolbox. Our approach is an enhancement to the finite state approach which makes the implementation of FSA approach efficient with respect to time and linguistic effort. The rest of the paper is organized as follows - section 2 is on related work. Design and architecture of the FSA based approach using corpus for paradigm mapping is presented in section 3. Experimental results are presented in section 4 and finally we conclude the paper with remarks on scope for future work.

2 Related Work

Attempts to map a word to a paradigm computationally have been attempted earlier. Rule based systems which map words to paradigms have been attempted (Sánchez-Cartagena et al., 2012), these systems use POS information or some additional user input from native language speakers to map words to paradigms instead of a corpus alone. Functional Morphology (M Forsberg, 2007) have been used to define morphology for language like Swedish and Finnish and Tools based on Functional Morphology namely Extract (M Forsberg, 2006) which suggest new words for lexicon and map them to paradigms have been developed. Functional Morphology based tools use constraint grammars to map words correctly to paradigms. The morphology of the language has to be fitted into the Functional Morphology definition to be able to use a tool like extract. Our work is close to the paradigm selection by Linden (Lindén, 2009) which is implemented to choose the appropriate paradigm for a guesser program for unknown words. Our paradigm selector method is implemented for Konkani verbs to quicken the process of building a verb MA.

We have defined our own paradigm structure for Konkani which uses FSA based sequencing of morphemes and suffix classes which provides a compact way to define a paradigm. This feature of grouping suffixes into classes such that if one suffix in a class gets attached to the stem then all the suffixes of the same class can be attached to the same stem gives a convenient and compact method to define a paradigm.

3 Design and Architecture of FSA based approach using corpus for paradigm mapping

Generally a verb has a stem to which suffixes are attached. Ambiguity in Konkani verb stem formation which prompted us to have a relevant paradigm design are –

- Given ending characters of verb citation forms there is no single rule to obtain a stem. For example if verb citation form ends with वप[1] (vaph; ; ending characters of verb citation form) as in case of धांवप (dhavap; to run; citation form of verb run), three rules can be applied. This gives rise to possible stems set with three entries namely {धांव, धांय, धां} ({dhav, dhayh, dha}; {run, not defined, not defined}; stems generated for verb run) of which only धांव (dhav; run; stem of verb run) is the correct stem. This is an ambiguity in stem formation for verbs. Hence simple rules based on verb citation form endings cannot be written to map verb citation forms to paradigms. Here a corpus is required to obtain support for the correct stem + suffix combination.
- A single verb citation form could give rise to more than one valid stem. Stems generated are such that two different stems of a single verb do not get attached to the same set of suffixes. For example verb आफडप (Aaphdaph; to touch; citation form of verb touch) gives rise to two stems namely आफुड (Aaphuda; touch; stem of verb touch) and आफड (Aaphd; ; stem of verb touch which has to be followed by appropriate suffix). An observation in such cases is that only one of these stems can exist as a valid verb form independently while the other stem has to be followed by some suffix and cannot appear as an independent verb form.
- Another case where there are more than one stem for a single verb citation form is when stems generated are alternatives to each other and one can be replaced by the other. Here unlike the above case, two stems of a single verb get attached with same set of suffixes. For example verb गावप (gavaph; to sing; citation form of verb sing) gives rise to two stems namely गाय (gayh; sing; stem of verb sing) and गा(ga; sing; stem of verb sing).

The Verb Morphological Analyzer has two main modules. It uses five resources as input and generates as output one crucial resource. Architecture of the FSA based Verb Morphological Analyzer using corpus for paradigm mapping is shown in FIGURE 1.

[1] A word in Konkani language is followed by transliteration in Roman Script, translation in English and gloss in brackets

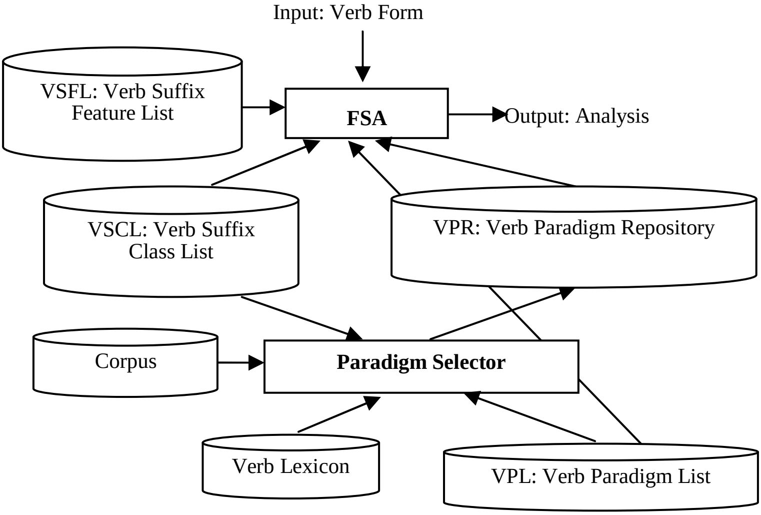

FIGURE 1 - Architecture of Verb Morphological Analyzer

The FSA module is used to sequence morphemes. It is used for both analysis and generation of word forms. Different verb forms in Konkani can be formed by attaching suffixes to stems of verbs (Sardessai, 1986). FIGURE 2 next shows the FSA constructed for Konkani Verb Analysis**.**

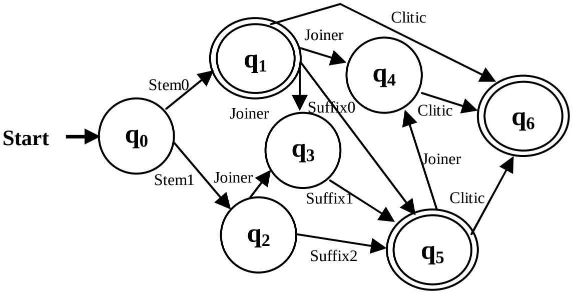

FIGURE 2 – FSA for Morphology of Konkani Verb

The other module is the Paradigm Selector module which maps a paradigm for an input verb and generates the crucial VPR resource. This module is run only when verbs not present in VPR are encountered.

The resources used in Verb Morphology Analyzer are -

- Verb lexicon: It contains citation form of a verb and Part of Speech category associated. This resource can be generated using the WordNet for the language.
- Corpus: Any standard corpus available in the language or a corpus generated by using any online newspaper of the language.
- Verb Suffix Feature List (VSFL): Verb Suffix Feature list has verb suffixes along with its associated morphological features.
- Verb Suffix Class List (VSCL): Here suffixes are grouped to form classes such that if one member of the class can be attached to a stem all other members of the same class can also be attached to the same stem.

- Verb Paradigm List (VPL): VPL has verb paradigms. It uses VSCL. We keep the following details of paradigm -
  - Paradigm-id: A unique identifier for each paradigm.
  - Root-Verb : A sample citation form of a verb which follows the paradigm
  - Stem-rule: A rule to obtain the stem for the given word. For example **delete,end,character#add,end,null** is a sample rule.
  - Suffix: A list of suffix classes, each class is followed by joiner property. Joiner property is used to accommodate changes which take place at morpheme boundary when two morphemes combine.

A word can have more than one stem in which case the stem-rule and suffix get repeated for each stem. The Paradigm Selector generates the Verb Paradigm Repository (VPR). VPR has an entire list of verbs in the language with the corresponding paradigm identifier of the paradigm it follows.

Algorithm: For every entry in verb lexicon, Paradigm Selector uses VPL Stem-rule to check compatibility of the verb with the paradigms. A verb may be compatible with more than one paradigm as the Stem-rule is same for those paradigms. The correct corresponding paradigm needs to be chosen from the compatible paradigm set. This is done with the help of a corpus. Assuming that each compatible paradigm is a valid paradigm, word variants are generated for that paradigm with the help of the FSA. The new generated words are then searched in the corpus. The paradigm corresponding to which maximum word variants are found is the correct paradigm corresponding to that word. Corresponding paradigm-id and sample paradigm, forms an entry in the generated output resource VPR.

4 Experimental Results

The implementation of the FSA based verb MA tool is done in Java using NetBeans IDE 6.9 on a Windows XP platform.

4.1 Results for Paradigm Selector Module

Data sets used by Paradigm Selector Module were the Konkani WordNet, the Asmitai corpus consisting of approximately 268,000 words, Verb Paradigm List and Verb Suffix Class List which were manually prepared. Experimental results for Paradigm Selector module are -

- Total number of Verbs used for the study after pruning the compound verbs was 1226.
- Total number of verbs for which paradigm were mapped by the program: 1003
  - Total number for which ambiguous paradigms (more than one) were mapped by the program: 159
  - Total number for which single paradigm were found by the program: 844
- Total number of verbs for which paradigms were not mapped by the program: 223
- Total number of verbs for which correct paradigms were found which was manually checked by linguist (true positives): 791
- Total number of verbs for which wrong paradigms were found which was manually checked by linguist (false positives): 53
- Total number of verbs for which ambiguous paradigms were found or no paradigm were found by the program (false negatives): 159+223 = 382

Precision = 0.93 Recall = 0.67 F-Score = 0.78

4.2 Results for FSA Module

The verbs for which no paradigm was selected were manually assigned paradigms. Verbs for which wrong paradigm was selected were corrected. This gave us an updated Verb-Paradigm Repository which was used by the FSA module.

Resources used as input was the tagged health and tourism domain corpus of ILCI used to obtain the verb variants, Verb Paradigm Repository generated, Verb Paradigm List, Verb Suffix Feature List and Verb Suffix Class List which were manually prepared. The corpus was partially validated at time of use.

Total number of unique verb variant forms used for the study was 8697 obtained from ILCI corpus by choosing the words tagged as verbs. Following results were obtained by the program -

- Total number of verb variants for which analysis was obtained: 7237
- Total number of verb variants for which analysis was not obtained: 1460

The verb variant for which analysis was not obtained was checked manually. We found the following three categories amongst the verb variants for which analysis was not obtained -

- Number of verb variants whose citation form was not present in the Lexicon(obtained from WordNet) thus was not present in Word Paradigm Repository: 632
- Number of verb variants which were tagged wrongly as verbs: 341
- Number of verb variants which were spelt wrongly in the corpus: 487

- Total number of verb variants for which correct analysis was obtained verified manually (true positives): 7183
- Total number of verb variants for which wrong analysis was obtained verified manually (false positives): 54
- Total number of verbs variants for which no analysis was obtained due to absence of citation form in Lexicon verified manually (false negatives): 632
- Total number of verb variants for which analysis was not obtained due to wrong tagging + wrong spelling verified manually (true negatives): 341+487 = 828

Precision =0.992 Recall = 0.919 F-Score = 0.954

Conclusion and Perspectives

From the results obtained for paradigm selector module we can say that the corpus is a reasonably good source to select paradigms for verbs. Human resource building effort can be substantially reduced with the use of this method. If the corpus is augmented with more forms of the verb then it would improve the recall of the paradigm selector module. This method can also be applied to the other grammatical categories such as nouns, adverbs etc.

Precision of FSA module for verbs is good. Tagged corpus when used to test the efficiency of the FSA module, results in more paradigm discovery and enhancement of the suffix list. It also suggests a list of verb citation forms which were absent in the lexicon. In addition it identifies spelling errors and tagging errors if any in the tagged corpus and could be used to validate the quality of the tagged corpus. The recall of FSA module can be improved by adding the citation forms which are not found in the lexicon to the VPR.

References

Amba Kulkarni, G UmaMaheshwar Rao, Building Morphological Analyzers and Generators for Indian Languages using FST, Tutorial for ICON 2010

D. Freitag, “Morphology induction from term clusters,” In Proceedings of the ninth conference on computational natural language learning (CoNLL), pp. 128–135, 2005.

Goldsmith, John and Aris Xanthos, Learning phonological categories. Language, 85(1):4–38, 2009.

Harald Hammarström, Lars Borin,Unsupervised Learning of Morphology, Computational Linguistics June 2011, Vol. 37, No. 2: 309–350

Harris, Zellig S., Phoneme to morpheme. Language, 31(2):190–222, 1955.

John Goldsmith, Linguistica: An Automatic Morphological Analyzer, CLS 36 [Papers from the 36th Meeting of the Chicago Linguistics Society]Volume1: The Main Session. 2000.

John Goldsmith, Unsupervised Learning of the, Morphology of a Natural Language, Computational Linguistics June 2001, Vol. 27, No. 2: 153–198.

Jurafsky, D., & Martin, J. H. Speech and Language Processing. Prentice-Hall, 2000

Lindén, K. and Tuovila, J., Corpus-based Paradigm Selection for Morphological Entries. In *Proceedings of NODALIDA 2009*, Odense, Denmark, May 2009

M. Forsberg H. Hammarström A. Ranta, Morphological Lexicon Extraction from Raw Text Data, LNAI 4139, pp.488-499,FinTAL 2006.

M. Forsberg and A. Ranta. Functional morphology. http://www.cs.chalmers.se/~markus/FM, 2007.

M. Porter, “An algorithm for suffix stripping program,” Vol. 14,pp. 130-137, 1980.

Madhavi Sardesai, Some Aspects of Konkani Grammar, M. Phil Thesis(1986).

Matthew Ameida, S.J, A Description of Konkani, Thomas Stephens Konknni Kendra, 1989

Suresh Jayvant Borkar, Konkani Vyakran, Konkani Bhasha Mandal, 1992.

Vícor M. Sánchez-Cartagena, Miquel Esplà-Gomis, Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Choosing the correct paradigm for unknown words in rule-based machine translation systems, Proceedings of the Third International Workshop on Free/Open-Source Rule-Based Machine Translation, , Gothenburg, Sweden, June 13-15, 2012.

Wicentowski, Richard. 2002. Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework. Ph.D. thesis, Johns Hopkins University, Baltimore, MD.

Xanthos, Aris, Yu Hu, and John Goldsmith, Exploring variant definitions of pointer length in MDL. In Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology at HLT-NAACL 2006.