International Journal on Emerging Technologies (Special Issue NCETST-2017) **8**(1): 62-64(2017)
(Published by Research Trend, Website: www.researchtrend.net*)*

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

A Survey of Applications of Finite Automata in Natural Language Processing

Raj Kishor Bisht
Department of Applied Sciences, Amrapali Institute of Technology and Sciences, Haldwani, (Uttarakhand), India

ABSTRACT: In the present paper a survey of applications of finite automata in natural language processing has been done. Applications to different natural language processing tasks like computational lexicography, morphological analysis etc. have been shown by considering some examples. Finally use of finite automata in Hindi tense recognition has been elaborated.

Keywords: Finite automata, Natural Language Processing, Morphology, Parsing.

I. INTRODUCTION

Natural language processing (NLP) comprises a number of tasks that deal with the processing of natural language through computers. The theory of automata plays a significant role in providing solutions of many problems in natural language processing. For example, speech recognition, spelling correction, information retrieval etc. Finite state methods are quite useful in processing natural language as the modeling of information using rules has many advantages for language modeling. Finite state automaton has a mathematical model which is quite understandable; data can be represented in a compacted form using finite state automaton and it allows automatic compilation of system components.

Here we will discuss some of the NLP tasks. Morphology is concerned with the study of the structure of a word and the pattern of word formation. A lexeme is a basic unit in a word that contains meaning. For example, 'walk', 'walks', 'walking', 'walked' have a common lexeme 'walk'. A morpheme is a minimal meaning bearing unit or grammatical function that is used to form a word. For example, the word 'walks' contains two morphemes; 'walk' and 's'. A morpheme is called free morpheme if it has independent existence as a single word and bound morpheme if it does not have independent existence as a single word, it must be attached to another word to get meaning. The morpheme 'walk' is free morpheme while the morpheme 's' is bound morpheme. A free morpheme can also be described as a stem and bound morpheme is described as an affix. An affix may be a prefix, or suffix (circumfix or infix also in some of the languages). Morphology can be further categorized in two ways: inflectional and derivational. A combination of a stem and an affix that produces a word of same class, that is, variation in tense, number etc. comes under inflectional morphology. For example, girl-girls, play-played etc. A combination of a stem and an affix that derives a word of new class, that is, which alters the meaning of the word comes under derivational morphology. For example, kind- kindness, computer-computerization etc. Inflections can further be categorized in two parts; regular and irregular. Inflections that follow a regular pattern in converting form singular to plural, present to past etc. are called regular inflections. For example, boy-boys, girl-girls, play-played etc.. Inflections that do not have a regular pattern for such conversion are called irregular. For example, man- men, go-went etc.

Finite state automata (deterministic and non deterministic finite automata) provide decision regarding acceptance and rejection of a string while transducers provide some output for a given input. Thus the two machines are quite useful in language processing tasks. Finite state automata are useful in deciding whether a given word belongs to a particular language or not. Similarly, transducers are useful in parsing and generation of words from their lexical form.

In the literature, we found a number of research papers regarding the application of finite automata in natural language processing. Jayara, Kornai and Sakarovitch [1] discussed the progress of work done in the direction of finite state methods and models in natural language processing. Some early work describing morphological analysis through finite state machine can be seen in Kaplan and Kay [8], Koskenniemi [4]. Shrivastava and Bhattachrayya [6] proposed HMM based part of speech tagging for Hindi. Sharma and Paul [7] developed a system identification and classification of clauses in Hindi text. Kumar, Deng and Byrene [9] presented a weighted finite state

transducer translation template model for statistical machine translation. Manning and Schutze [2] and Jurafsky and Martin [3] provided details of applications of finite automata in NLP in their books.

II. FINITE AUTOMATA IN NLP

In this section a survey of applications of finite automata in natural language processing has been made.

A. Language Recognizer

There are many tasks that need language recognizing mechanism. For example, spelling checker, morphological analysis, language identification etc.. Finite state machine are quite useful as a language recognizer. For a given word, a NFA can be designed easily that recognize the word. For example, NFA for the words 'boy' and 'bat' is shown in the Fig 1. Similarly for every word a NFA can be designed and the different NFA's can be combined to form spelling checker or dictionary compilation for a language. Mohri [ ] showed the application of finite automata in large scale dictionary compilation and indexation of natural language text.

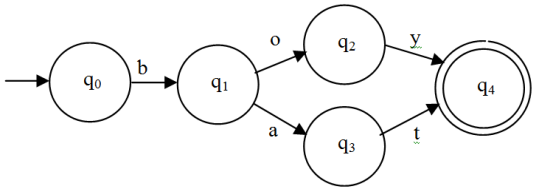

Fig. 1. NFA for the words 'boy' and 'bat'.

Inflectional morphology can be well recognized by finite automata. For each category of words we can form a separate NFA and then combine them using λ transitions. For example, nouns and their plural can be recognized through one NFA and verbs and their different forms can be recognized through another NFA and finally the two NFA can be combined. Figure 2 shows the NFA for some words and their morphological variations.

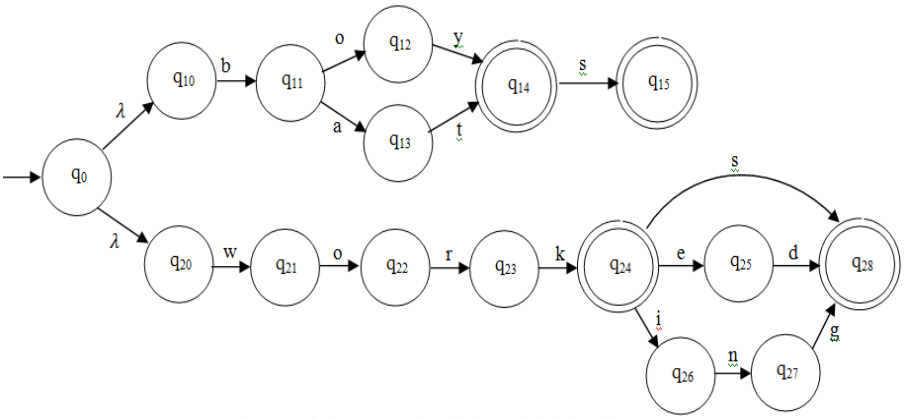

Fig. 2. NFA for the some words and their morphological variations.

B. Morphological parsing and generation

Morphological parsing is the process of producing a lexical structure of a word, that is, breaking a word into stem and affixes and labeling the morphemes with category label. For example the word 'books' can be parsed as book+s and further as book + N+PL, the word 'went' can be parsed as go+V+PAST etc. generation is the reverse process of parsing, that is combining the lexical form of a word to produce the word. For example, box+N+Pl generates the word 'boxes'. Finite state transducers are quite useful in morphological parsing. Let us consider the lexicon form of regular inflectional 'girl +N +PL'. Fig. 3 shows the transducer which convert the lexicon form 'girl+N+PL' into the word 'girls'. Let us assume that *x* represents the word 'girl' for simplification purpose as for every regular singular noun the input and output will remain same in a transducer, that's why a variable *x* can be used for a noun. The word 'girl' can be replaced by any other regular noun like 'boy'.

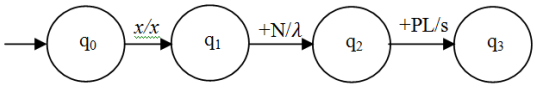

Fig. 3. Generation of word from its lexical form through transducer.

Here simple regular nouns have been considered on exemplarily basis, orthographic rules can be applied for irregular inflections and derivations and NFAs can be designed for every word and its variations in a certain language.

C. Tense recognition in Hindi

Tenses in Hindi are important part of grammar. Their knowledge is quite useful in translation from Hindi to another language. Here we will show the application of finite state machines in automatic recognition of tenses in Hindi language. We have some grammatical rules in Hindi through which we can identify the tense of a sentence and translate the sentence in English. We know that there are three tenses and each tense has a four sub tenses. For example, the sentences which end with 'Rkk gS] rh gS] rs gS' etc. come under present indefinite tense, like *eksgu QqVcky [ksyrk gS^ and the sentence which end with 'jgk gS] jgh gS] jgs gS'' etcetc. come under present continuous tense, like *xhrk [ksy jgh gS^. We may have the following parse tree structure of the second Hindi sentence for tense recognition purpose. Here the non terminal EF stands for 'End Form', other non terminal have usual meaning.

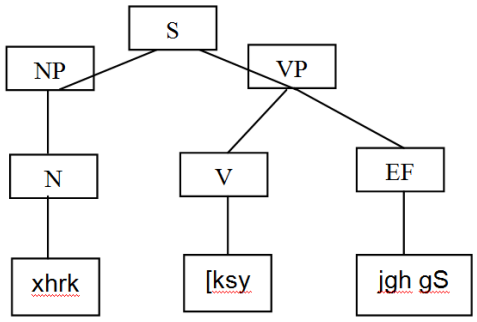

Fig. 4. Parse tree for the sentence *xhrk [ksy jgh gS^.

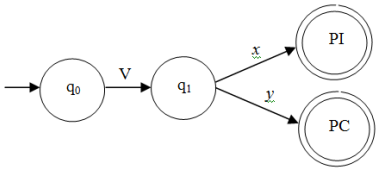

Fig. 5. NFA for recognition of two tenses.

A finite state machine can be designed for the verb phrase that appears at the end of the sentence for deciding the tense of the sentence. Tough there are number of complexities and variations in Hindi sentences but here a simple example has been taken to introduce the concept. Fig. 5 shows the recognition of two tenses in Hindi sentences through NFA. The abbreviations showing final states are as follows: PI-Present indefinite and PC-Present continuous. Further $x \in$ {Rkk gS] rh gS] rs gS} , $y \in$ { jgk gS] jgh gS] jgs gS},

III. CONCLUSION

In the present paper a survey of applications of finite automata in natural language processing has been done. Some examples have been taken to show the morphological analysis and parsing of English sentences. Finally the application of finite automata has been shown in tense recognition of Hindi sentences which is an important part of rule based machine translation.

REFERENCES

[1]. A. Jayara, A. Kornai and A.Sakarovitch, "Finite-state methods and models in natural language processing, *Natural Language Engineering* **17** (2): 141–144, 2011.

[2]. C.D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing. MIT Press, 2002.

[3]. D. Jurafsky, and J.H. Martin, Speech and Language Processing, Pearson Eucation India, 2004.

[4]. K. Koskenniemi, "Finite state morphology and and information retrieval" in the proceedings of the ECAI workshop extended finite state models of language, 1996, pp. 42-45.

[5]. M. Mohri, "On some applications of finite automata theory to natural language processing", *Natural Language Engineering*, Vol. **1**(1), pp. 000-000, 1995.

[6]. M. Shrivastava, P. Bhattachrayya, "Hindi POS Tagger Using Naive Stemming : Harnessing Morphological Information Without Extensive Linguistic Knowledge, in the proceedings of ICON-2008: 6th International Conference on Natural Language Processing, Macmillan Publishers, India.

[7]. R. Sharma, S.Paul, "A rule based approach for automatic clause boundary detection and classification in Hindi", Proceedings of the 5th Workshop on South and Southeast Asian NLP, *25th International Conference on Computational Linguistics*, pp. 102–111, 2014.

[8]. R.M.Kaplan, M. Kay, "Regular models of rule syatems", *Computational Linguistics,* vol. **20**(3), pp. 331-378, 1994.

[9]. S. Kumar, Y. Deng and W. Byrne, "A weighted finite state transducer translation template model for statistical machine translation", *Natural Language Engineering* **12** (1): 35–75, 2005.