Automatic Question Generator in Tamil

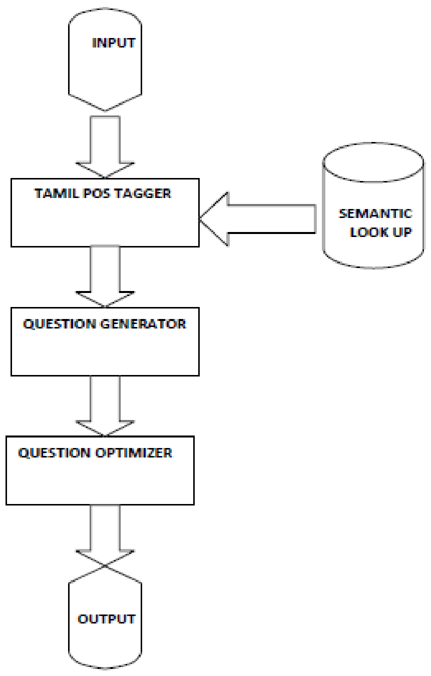
[1]N.Vignesh [2]S.Sowmya

1. Research Associate, Indian Institute of Management Ahmedabad,
2. SDE, ACS Oracle India,

Abstract:

The automatic question generator for Tamil is a Language processing system which is used to generate questions for valid Tamil sentences which follows the grammatical rules and constraints imposed by the language. This Natural Language Processing (NLP) system combines the knowledge about the language and application domain to automatically generate questions for Tamil sentences. Even though this language processing system dynamically generates optimal questions for sentences, this type of system requires lot of information for processing data. Structural correctness of the input sentence and proper semantic analysis leads to good results wherein misunderstanding and lagging of semantic information of resources may lead to uncontrolled output.

Keywords: NLP, dynamical, optimal questions



1. INTRODUCTION

Natural language generation (NLG) systems combine knowledge about language and the application domain to automatically produce documents, reports, explanations, help messages, and other kinds of texts. This automatic Tamil Question Generator is a form of NLG which produces optimal question set for any given Tamil sentence provided the input is according to the Tamil Grammar.

The figure below describes the architecture of the Question Generator system.

This paper organized as section 2 explains the state of the are natural language generation, section 3 explains the proposed natural language Tamil question generation system, section 4 describes the algorithm of the proposed system and the final section concludes with the scope of this work.

2. EXISTING SYSTEM

Automatic generation of natural language text is an essential task in many language processing like summarization, automatic document generation, question answering system, translator etc. It consists of a learner to learn how to realize a sentence from the content of semantic role information. This learner is to be designed as a statistical model that is formulated from a preprocessed corpus of sentences. This preprocessing is handled by pos tagging, chunking and semantic role labeling. A POS tagging tool [1] for Tamil is developed in Anna University. Similarly Chunking tool [2] for Tamil is also developed at Anna University which can also been used for Chunking of POS tagged texts. These systems serve to be the pre runner/ existing works that comply with our project.

3. PROPOSED SYSTEM

3.1 INPUT:

The input to this language processing system is a valid Tamil sentence which obeys all the structural and grammatical constraints of that language. For example the input may be a sentence of this kind:

நேற்று அழகிய ராமன் பழகிய சீதையை தேடி காட்டுக்கு சென்றான்.

3.2 TAMIL POS TAGGER:

This part of the language processing system tags each of the words in the given input sentence with suitable descriptor tags which enables the generation of questions automatically with the help of those tags. The process of tagging and the tags used are described below.

3.2.1 NOUN MARKER TAG:

In the process of tagging the noun marker each and every word of the sentence is tagged as the noun initially and after completion of the tagging procedure by matching the semantic and structural rules of the language the remaining untagged words are considered to be noun. This method is followed since there is no particular structural rule for noun detection.

Therefore in the above example the words "நேற்று, அழகிய, ராமன் பழகிய, சீதையை, தேடி, காட்டுக்கு, சென்றான்" gets tagged as noun initially.

3.2.2 VERB MARKER TAG:

In the process of tagging the verb marker the following sub tags are also made thus enabling the generation of question which is yet to happen in the later phase. The sub tagging done which aids the processes of verb identification are described below.

3.2.2.1 GENDER MARKER TAG:

Each word of the input Tamil sentence is checked for the presence of Gender markers such as ஆண், ஆள், கள், அது at the end of the word which describes male, female, plural and genders other than male or female .If such gender marker tags are found in the word then the algorithm proceeds in finding the tense markers which is described below.

3.2.2.2 TENSE MARKER TAG:

A tense marker appears just beneath the gender marker and is used to conclude that the word with a tense marker preceded by a gender marker is definitely a verb. The various tense markers are:

Present tense: கிறு, கின்று, ஆநின்று

Past tense: த், ட், ற், இன்

Future tense: ப், வ்

Thus a word is tagged as a verb if it is of the form:
Verb=action word + tense marker + gender marker

சென்றான் = செல் + ற் + ஆண்

Thus in this example the word "சென்றான்" gets tagged as "past tense male "verb marker.

3.2.3 TIME MARKER TAG:

The word is checked against the following predefined time marker words such as நேற்று, இன்று, நாளை and some other date markers such as date and week markers and if these marker words are present the word gets tagged with a time marker descriptor tag. In the main example given about the word "நேற்று" gets a Time marker tag.

3.2.4 NOUN DESCRIPTOR TAG:

For tagging a word as a noun descriptor the word has to be checked against these following rules. The word must end with the rhyme "அ" and word which succeeds this word must be a noun

Considering the above main example:
அழகிய ராமன்'– The word "ராமன்" already got tagged as noun marker by rule 3.2.1 and the word 'அழகிய= அழகு + ய் + அ 'ends with the rhyme "அ". As the two rules are satisfied this word "அழகிய"gets tagged as the noun descriptor. Similarly the word "பழகிய= பழகு + ய் + அ" which obeys the above two rules gets tagged as noun descriptor or noun quantifier.

3.2.5 VERB DESCRIPTOR TAG:

The word which is to be tagged as a verb descriptor must end with the rhyme either "அ" or '"இ"'. By checking against this rule the word can be tagged as the verb descriptor. In the above given example considering the word "தேடி= தேடு + இ" ends with the rhyme '"இ"' thus satisfying the above rule and hence this word gets tagged as the verb quantifier or verb descriptor tag. Thus coming across this tagger section of the system tags all the words of the input sentence with appropriate taggers and sends this as input to the question generator phase.

Considering the above given example the whole sentence gets tagged as below:

நேற்று அழகிய ராமன் பழகிய சீதையை தேடி காட்டுக்கு சென்றான்.

Time Descriptor + Noun Quantifier + Noun + Noun Quantifier + Noun + Verb Quantifier + Noun + Past tense Male singular verb.

3.3 QUESTION GENERATOR:

The phase of the language processing system gets the tagged sentence as the input and produces all possible questions as the output. The appropriate tags are used to generate questions by following the rules given below The tagged sentence is again scanned from left to right word by word and each time a word is replaced in the sentence based on appropriate tags to that word to generate questions:

3.3.1 RULES FOR QUESTION GENERATION:

Rule 1: Replace the time marker tag in the sentence with the word "எப்பொழுது" to generate a question. Considering the example the question generated by replacing the time marker will be of the form:
எப்பொழுது அழகிய ராமன் பழகிய சீதையை தேடி காட்டுக்கு சென்றான் ?

Rule 2: Replace the noun quantifier tag in the sentence with the word "எத்தகைய" to generate a question.

Rule 3: Replace the first occurrence of noun tag in the sentence with the word "யார்" to generate a question if the gender is either male or female else replace the word with "எது". Considering the example the question generated by replacing the time marker will be of the form:

நேற்று அழகிய யார் பழகிய சீதையை தேடி காட்டுக்கு சென்றான்?

Rule 4: Replace the verb quantifier tag in the sentence with the word "எதற்கு" and also delete the noun and its corresponding noun quantifier to generate a question. Considering the example the question generated by replacing the time marker will be of the form:

நேற்று அழகிய ராமன் எதற்கு காட்டுக்கு சென்றான் ?

Rule 5: Find the words tagged as nouns and ending with the rhymes 'ஐ, ஆள், க்கு, இன், அது, கண்" and replace them with 'யாரை யாரால் யாருக்கு யாரின் யாரது யார்கண்' if the verb in that sentence is either has either male or female gender descriptor else replace them with 'எதை எதால் எதற்கு எதின்'.

Note: The second question generated by following the rule 5 is not valid which can be detected by checking the structural pattern of the language. (Check 5.Future work portions for further reference regarding this issue)

Thus the various questions generated by following the above rules for the given input sentence are as follows:

1) எப்பொழுது அழகிய ராமன் பழகிய சீதையை தேடி காட்டுக்கு சென்றான் ?
2) நேற்று எத்தகைய ராமன் எத்தகைய சீதையை தேடி காட்டுக்கு சென்றான் ?
3) நேற்று அழகிய யார் பழகிய சீதையை தேடி காட்டுக்கு சென்றான் ?
4) நேற்று அழகிய ராமன் எதற்கு காட்டுக்கு சென்றான் ?
5) நேற்று அழகிய ராமன் பழகிய யாரை தேடி காட்டுக்கு சென்றான் ?
6) நேற்று அழகிய ராமன் பழகிய சீதையை தேடி யாருக்கு சென்றான் ?

3.4 QUESTION OPTIMIZER

This phase of the question generator system is used to narrow down to a single optimal question for a given sentence rather than set of all possible questions. This can be done by following the below described rules:

Allot points to each tag based of precedence of tags say:

TAG	POINTS
Verb Quantifier	10
Time Quantifier	9
Noun	8
Noun Quantifier	7
Verb	5

By using the above precedence check each replacement and choose the question with maximum points as the optimal one. Considering the above approach the optimal question that is generated from the possible set of questions is:

அழகிய ராமன் எதற்கு காட்டுக்கு சென்றான் ?

4. ALGORITHM

```
Algo QuestionGen()

begin
    nounCount=0;

    While(isNextWord()!=NULL) do
    begin
    // tamilPosTagging()
        tag each word as noun;
        if( isGenderDescriptor() and isTenseDescriptor()) then
            tag word as verb;
        else if(isRhymeEnd("அ") and isNextWord()=noun) then
            tag word as noun quantifier;
        else if(isRhymeEnd("அ") or isRhymeEnd("இ") then
            tag word as verb quantifier;
        else if(isTimeDescriptorsAvailable()) then
            tag word as time descriptor;
    end

    While(isNextWord()!=NULL) do
        begin
        //questionGeneration()
            if(IsNounTag() and nounCount=0) then
                if(isMale() or isFemale()) then
                    replace word with "யார்";
                else
                    replace word with "எது";
                nounCount++;
            if(isTimeTag()) then
                    replace word with "எப்பொழுது" ;
            if(isNounQuantifier())then
                    replace word with "எத்தகைய";
            if(isVerbQuantifier()) then
                    replace word with "எதற்கு"
                    remove(nounQuantifier(prevWord()));
                    remove (prevWord());

        end

end
```

5. FUTURE WORK

This question generation system which we have designed allows replacing only a word in the input sentence with one question word but it follows the structural and grammatical correctness of the language. This efficient question generation may be improved by allowing more than one replacements of tagged words in a sentence with question words and generating optimal questions by keeping an upper and lower boundary on number of replacements permitted. Also classification of nouns based on living and non living things helps us to better understand the noun quantifier and additional information can be provided to the noun descriptor and what it currently quantifies can be identified and generation of corresponding questions can be made possible.

6. CONCLUSION

Thus this automatic question generation system implemented above can be made use for framing questions given a valid input sentence or group of sentences. This language processing system finds application in generating optimal questions which aids in better understanding of structural and grammatical aspects of a language. This same technique can be applied to other languages only varying the basic blocks of grammar for that corresponding language.

7. ACKNOWLEGEMENT

I express my deep gratitude to my guide, **Dr. Rajeshwari Sridhar**, College of Engineering Guindy, Anna University, Chennai for guiding me through every phase of this project. I appreciate her thoroughness, tolerance and ability to share her knowledge with me. I thank her for being easily approachable and quite thoughtful. Apart from adding her own input, she has encouraged me to think on my own and give form to my thoughts. I owe her for harnessing my potential and bringing out the best in me. Without her immense support through every step of the way, I could never have it to this extent.

8. REFERENCES

[1] S.Laksmana Pandian, T.V.Geetha, Morpheme based Language Model for Part of Speech tagging, Research journal "POLIBITS" on Computer science and computer engineering with applications Issue 38 (July-December 2008)

[2] S. Lakshmana Pandian, T. V. Geetha: CRF based Tamil part of speech tagging and chunging, Lecture Notes in Computer Science, 2009(5221).

[3] S.Lakshmana Pandiyan and T.V.Geetha “Morpheme based language model for Tamil Part-of-Speech tagging” in the International Journal of Communications and Engineering

[4] S.Lakshmana Pandiyan and T.V.Geetha. “Semantic role based Tamil sentence generator” in 2009 International Conference on Asian Languages Processing.

[5] S.Lakshmana Pandiyan and T.V.Geetha, ”Semantic role labelling for Tamil documents” in International Journal of Recent Trends in Engineering, Vol. 1, No.1, May 2009.

N.Vignesh is working as a Research Associate at Indian Institute of Management, Ahmedabad. He received his B.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Chennai in 2013. His research interest includes Natural Language Processing and Computer Networks.

S.Sowmya is working as a Software Development Engineer at ACS, Oracle India. She received her B.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Chennai in 2013. Her research interest includes Natural Language Processing and Computer Networks.