உத்தமம்
INFITT
உலகத் தமிழ்த் தகவல் தொழில்நுட்ப மன்றம்
International Forum for Information Technology in Tamil

Penn
University of Pennsylvania

மாநாட்டுக் கட்டுரைகள்
CONFERENCE PAPERS

TAMIL INTERNET 2011

தமிழ் இணையம் 2011

Animated Story Visualizer for Tamil Text

M. Janani, Dr. T. Mala
Department of Information Science and Technology
Anna University, Chennai, India
janani_mrl@yahoo.com, malanehru@annauniv.edu

Abstract

Natural language is a straightforward and efficient medium for describing visual facts and mental images. The System uses a novel approach to generate animation from Tamil texts such as stories. Tamil text is pre-processed and the necessary features like named entities, environmental constraints, temporal and emotion constraints for the given stories are extracted and placed in the database. The system automatically generates a query based on the users input and compare it with features stored in the database. Finally animation is dynamically generated using an external motion synthesis system. Using this system, even greenhorn users can generate animation quickly and easily by giving the Tamil text.

***Keywords*— Computer animation, Natural Language Processing, Pre-processing, Feature Extraction, Motion synthesis**

I. INTRODUCTION

These days, animations are widely used in many applications, such as cartoons, web graphics, games, and so on. Computer animation is one of the best methods for depicting the dynamic content. A medium is necessary for the animation to be created in a convenient and natural manner. It should be possible to describe the scenes directly from natural language.NLP is an easy and effective way to analyze, understand and generate languages that humans use naturally.

The aim of this work is to generate an animation from Tamil texts such as movie scripts or stories. Training input text is given to the pre-processing module. Here tokenization is performed and the tokens are given to the morphological analyser which is used to convert the tokens into a POS tags. Information related to named entities, temporal constraints, emotion and environment inference features are extracted. A query is generated automatically from the input text which contains information for the search process and compares it with the information already stored in the database. Finally motion synthesis generates an animation. The interactions between characters are handled by this module based on the information provided in the database.

II. RELATED WORK

Generating animation from natural language texts has been a challenge. WordsEye developed by Coyne and Sproat [1] converts natural language texts to a scene. WordsEye focuses on generating a still image, when a character motion is specified in a given text, the system simply prefer to pose for

the action generated from the database. The Carsim system [2] describes a new version of text-to-scene converter that handles texts describing car accidents using computer program and it is visualized in the 3D environment. Storytelling System [3] illustrates a system called Interactive e-Hon, which provides storytelling in the form of animation and conversation translated from original text. A Constraint based scene conversion system [4] describes a Text2Scene conversion method which automatically converts text into 3D scenes. A large database of 3D models is used by this method to depict entities and actions.

III. SYSTEM OVERVIEW

In this section, overview of our system (Figure 1) is given, where the major components are identified. When the Tamil text is given to the system, Tamil text is pre-processed and the information are extracted and stored in the database along with the objects created. When an input text is given to the system it automatically generates the query from the input text and compares it with the information stored in the database. An animation is then generated using an external motion synthesis system.

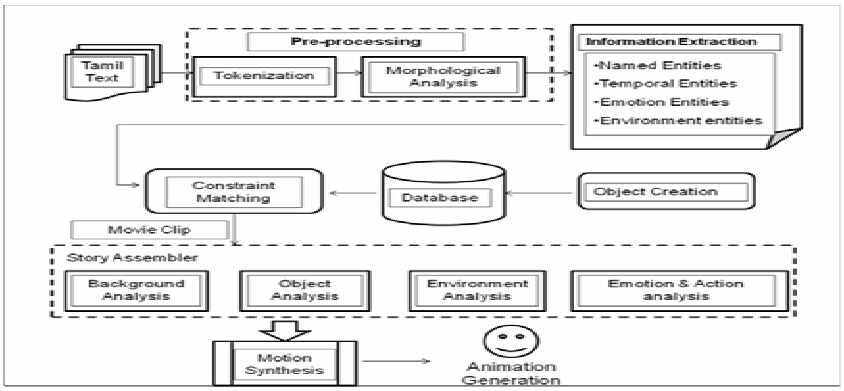

Figure 1. SYSTEM OVERVIEW

PRE-PROCESSING

When the Tamil text is given to the system, natural language processes (Tokenization and morphological analysis) are applied first.

Tokenization

The first step in NLP is to identify tokens, which decomposes the delimiters like punctuation and whitespaces. Here Tamil text is given as the input to the tokenizer which breaks the text into meaningful tokens. The tokens generated by the tokenizer are passed to the analysis engine.

Morphological Analysis

RCILTS [5] developed a tool called Atcharam, an analyser which performs Morphological Analysis for Tamil text. The Morphological analyser takes a derived word as input and separates it into root word and associated morphemes. It is the basic tool used in spell checker, grammar checker, parser and machine translation systems. It has two major modules noun analyzer and verb analyzer.

வண்டுகளுக்கு	வண்டு< noun >
	கள்< plural >
	உக்கு<case marker >

படித்தான்	படி< verb >
	த்த்< past tense marker>
	ன்< gender >

Tamil Noun and Verb classification example

By this method morphemes are generated and given to the learning process where the necessary informations are extracted.

INFORMATION EXTRACTION

OBJECT IDENTIFIERS

Object Identifiers recognize named entities in text by Named Entity Recognition (NER). "Rule based approach" is used to extract named entities from the given text. Initially, root words say Noun, verb, adjective, pronoun, adverb from text file are extracted. Rules are created based on prefix and postfix of noun, i.e. noun that occurs between verb and noun, noun that occurs between noun and noun, noun that occurs between noun and verb and so on. If any of the above rules satisfies the input text named entities are extracted. Here is an example,

Input: ஒரு குளத்தில் எறும்பு தத்தளித்து

Given input text is pre-processed and the root words are extracted.

குளம்<Noun>
எறும்பு<Noun>
தத்தளி<Verb>

Now the rules are applied to this extracted root words. Here எறும்பு comes between noun and verb which satisfies the rule is extracted.

TEMPORAL AND EMOTION EXTRACTION

Temporal reasoning in NLP involves extraction, representation and reasoning with time and events in the natural language text. Here to extract temporal constraints, "manually created dictionary" is used. The root words are compared with the manually created dictionary and temporal constraints are extracted if the input text satisfies the inferences present in the dictionary. Similarly different emotion present in the text is also extracted using manually created dictionary.

Figure 2 shows the different emotional constraints to be depicted.

HAPPY	இன்பம்,மகிழ்ச்சி,குதூகலம்,சந்தோஷம்,ஆனந்தம்,களிப்பு,பெருமிதம்,சிரிப்பு
ANGER	அகங்காரம்,சினம்,கோபம்,வெறுப்பு,எரிச்சல்,சீற்றம் தாபம்
SURPRISE	அதிசயம்,ஆச்சரியம், பிரமிப்பு,மலைப்பு, வியப்பு
FEAR	அச்சம்,பயம்,பீர்,பொருமல்,விதிர்ப்பு,கவலை,அஞ்சு,கலக்கம்,நடுக்கம்
SADNESS	சோகம்,அழுகை,சோர்ந்த, துயரம்,வருத்தம், துன்பம், கண்ணீர், கூச்சல், அலறு, கதறு,புலம்பு,கத்து, முழக்கம்,கூக்குரல், துக்கம், வாட்டம், விசனம்

Figure 2 Emotion Constraints

Finally, environmental constraints that specify location and actions are extracted.

ANIMATION GENERATION

Movie clips for the extracted information are created using Adobe Flash professional and stored onto the database.

CONSTRAINT MATCHING

When an input text is given, information constraint should match with the information present in the database to generate animation. String matching algorithm is used to compare the information extracted and information stored in the database. Let P[1...M] and T[1...N] be the character array for the given string. Pattern *P* is said to occur with shift *s* in text *T.* To find all valid shifts or possible values of s so that *P[1..m]* = *T[s+1..s+m]* ; There are *n-m+1* possible values of *s*.

Procedure String Matcher(T,P)

1. $n \leftarrow$ length [*T*];
2. $m \leftarrow$ length[*P*];
3. for $s \leftarrow 0$ *to* $n-m$
4. do if *P[1..m]* = *T[s+1..s+m]*
5. then shift *s* is valid

Find first match of a pattern of length M in a text stream of length N.

The extraction of Pattern கா க ம் is done by,

காகம் M = 4

க ழு தை ஆ டு கா க ம் கு ர ங் கு

கா க ம்

கா க ம்

கா க ம்

கா க ம்

கா க ம்

கா க ம்

By this method exact string is matched from the database for the given information.

STORY ASSEMBLER

Storyboards are the only way to convey rich information, viewing a particular order of events in a most appealing way. Basically the system searches for noun and verb from the given input text then automatically assemble and analyse the subsequence like background, named entities, temporal, emotion and action movie clip from a database that matches the constraints.

MOTION SYNTHESIS

Animation is generated by motion synthesis by efficiently connecting the movie clips that are assembled by the story assembler from the database. The character and objects interactions are handled by this module based on the information that the movie clips have. Timeline specifies what

kind of action occurs at particular time. Once the timeline has been set, animation is generated for the given Tamil text. Figure 3 shows the animation generated for the given Tamil Text

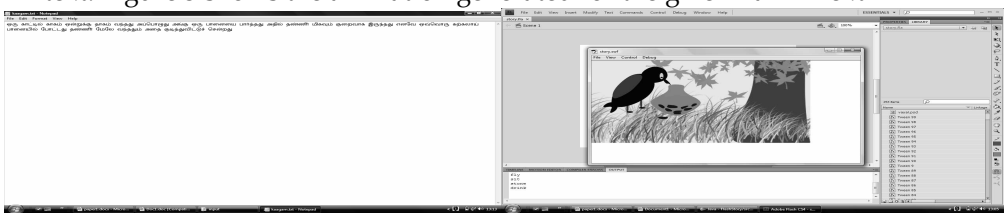

Figure 3 Animation is generated from Tamil text

PERFORMANCE ANALYSIS

The performance analysis is used to monitor the functioning, efficiency, accuracy and other such aspects of a system. For the analysis performed for the learning process, the overall accuracy obtained is 83%. Figure 4 shows the Performance analysis for Animation generated from the Tamil text.

	No.of Stories	Exactly generated	Approx. generated
viewer1	18	15	3
viewer2	18	13	5
viewer3	18	14	4
viewer4	18	16	2
viewer5	18	14	4

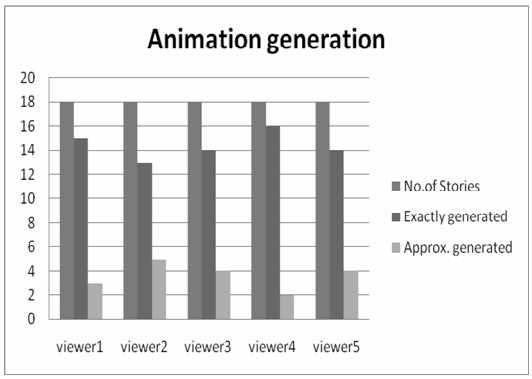

Table 1 Test case for Animation Generation *Figure 4 Performance analysis for Animation generation*

The overall accuracy obtained for the generation of animation is 80%. The performance can be further improved by generating rules and optimizing the learning process.

CONCLUSION AND FUTURE WORK

The system provides automated generation of animation from Tamil text which provides a new approach for users to create animation quickly. The proposed method takes Tamil text as the input and it is pre-processed and features like named entities, temporal constraints, emotion and environmental constraints are extracted and animation is generated dynamically by motion synthesis. Even non-professional people can rely on this system and they can generate animation quickly and easily by giving the Tamil text. Future work can be extended by generating animation via automatic speech recognition rather than text.

REFERENCES

- Coyne.B and Sproat.R, "*Wordseye: an automatic text-to-scene conversion system*", In Proceedings of SIGGRAPH 2001, pp. 487-496.
- Johansson.R, Berglund.A, Danielsson.M and Nugues.N, "*Automatic Text-to-Scene Conversion in the Traffic Accident Domain*", In Proceedings of The Nineteenth International Joint Conference on Artificial Intelligence, 2005, pp. 1073–1078.
- Kaoru Sumi, Mizue Nagata, "*Animated Storytelling System via Text*"**,** In Proceedings of SIGCHI International Conference on Advances in computer entertainment technology,2006.
- Liuzhou Wu and Zelin Chen,"*A Constraint-based Text-to-Scene Conversion System*", In Proceedings of International Conference on Computational Intelligence and Software Engineering 2009.
- Anandan, P., Ranjani Parthasarathy & Geetha, T.V., "*Morphological Analyser for Tamil*", ICON 2002, RCILTS-Tamil, Anna University, India.