A COMPUTATIONAL GRAMMAR OF SINHALA FOR ENGLISH-SINHALA MACHINE TRANSLATION

By

Budditha Hettige
(08/8021)

Supervised by
Prof. Asoka S. Karunananda
Overview

1. Introduction
2. Aim and Objectives
3. Approaches to Machine Translation
4. Overview of English and Sinhala Languages
5. A Novel Approach to Machine Translation
6. Design and Implementation
7. Evaluation
8. Conclusion & Further works
9. Demonstration
Introduction

• According to the latest estimates, Sri Lanka has a population of about 20 million
• Majority of Sri Lankans (74%) use Sinhala as the spoken and written language
• Sri Lankans also use English as a Second Language and many sources are in English
• However, only about 10% of the population can read and write English Language well
• **18 million** cannot read and write English Language perfectly
• This leads to what is called “Language barrier”
Language Barrier

- Barrier for communication among communities using different languages
- It affects on acquisition of world knowledge
- It also a barrier for discovery and dissemination of knowledge
Solutions for Language Barrier

• Learning English by the entire population, but this is not practical and also cannot neglect the power of the mother tongue in the context of knowledge.

• Usage of translators
  – Human Translation
  – Machine Translation

• Machine Translation has been a potential solution.

• Machine Translation is cost effective and faster than the human translation.

• Many Asian and European countries use machine translation systems to solve their language barrier.
Machine Translation

• Machine Translation is a computer software that translates text or voice from one natural language to another with or without human assistance [Wikipedia]

• This is an inherently difficult task due to diversification of natural languages

• As such many machine translation approaches appear to be rather ad-hoc

• Thus developing theoretical-based approaches to machine translation turn out to be a research challenge
Lack of theoretical based approach to machine translation has been one of the major reasons for the development of efficient computer based solutions for natural language translations
Aim

Design and develop an agent based English to Sinhala machine translation system with a theoretical basis
Objectives

• **Objective 1:**
  Critically review the existing systems for machine translation.

• **Objective 2:**
  Study the concepts / techniques for Natural Language Processing.

• **Objective 3:**
  Study the concepts/techniques and adapt existing Morphological Analyzers and Parsers for English language

• **Objective 4:**
  Design and develop Morphological Analyzer/Generator and Parser/Composer for Sinhala Language
Objectives (cont..)

- **Objective 5:**
  Design and develop lexical databases for English to Sinhala Machine Translation

- **Objective 6:**
  Design and develop English to Sinhala Machine Translation system by integrating above sub-systems to form a Multi agent system

- **Objective 7:**
  Evaluate the system
Hypothesis

Concepts of “Varanageema” (Conjugation) in Sinhala language can be used to drive English to Sinhala Machine translation
Approaches to Machine Translation

- Human-assisted
- Rule-based
- Statistical
- Example-based
- Knowledge-based
- Hybrid
- Agent-based
Human-assisted Approach

- Uses human interaction for the pre editing, post editing and/or intermediate editing stages

- Examples
  - Anusaaraka
    - Translates among Indian languages and English to Hindi
  - MaTra
    - Translates English to Hindi
    - Produce understandable output than wide coverage
Rule-based Approach

- Gives grammatically correct translation by using the set of rules

- Examples
  - Apertium
    - Open source system
    - Can be used to translate any related two languages
  - OpenLogos
    - Open source
    - English and German to major European languages
Statistical Approach

- Generates translations using statistical methods based on bilingual text corpora
- **Examples**
  - Babel Fish
    - Web based application develop by the AltaVista
    - Can translate among English French Greek etc.
  - Bing Translator
    - Translate text or entire web page into different languages
  - Google Translator
Example-based Approach

• Uses bilingual corpus with the partial text for the machine translation
• It uses translation memories
• Examples
  – OpenMaTrExis
    • Open source
    • Translates Basque to English
  – Kyoto-U
    • Translates English – Japanese
Knowledge-based Approach

- Uses knowledge for machine translation
- Extended idea of the example-based machine translation
- Uses linguistic and computational instructions, which are supplied by a human

Examples

- EDR (Electronic Dictionary Research)
  - English-Japanese
- KANT (Knowledge-based Accurate Natural-language Translation)
  - Translate French, German and Japanese
Hybrid Approach

• Combines two or more above approaches to machine translation

• Examples
  – SYSTEMS Enterprise server 7 by SYSTRAN
    • Powered by SYSTRAN's hybrid machine translation engine
    • It combines rule-based and statistical approaches
    • 52 Commercially available language pairs
  – TransEasy
    • Uses rule-based and statistical approaches
    • Translates Chinese to English
Agent-based Approach

- Translates through agents communication
- Examples
  - TALISMAN
    - Linguistic Agents are used - preprocessing, morphological analysis, syntactic analysis etc
  - SYSTRAN
  - Car insurance System by Majenta
    - Analyzing and classifying 25000 documents related to car insurance
    - Multi-agent based method is used for text understanding
Existing Approaches for English to Sinhala Machine Translation

- **OpenTM**
  - Example-based/Statistical
  - Suitable for Scripting languages

- **Vithanage’s English to Sinhala MT System**
  - Rule-based
  - Weather forecasting domain

- **Fernando and others English to Sinhala MT System**
  - Artificial Neural Networks method
  - Limited domain
Concepts and Techniques for Machine Translation

- Morphological analysis
- Syntax Analysis
- Semantic analysis
- Multi Agent Systems
Morphological Analysis (MA)

• The morphological analysis is the identification (analysis) of the structure of morphemes

• **First attempt** - Indian linguist Panini formulated the 3,959 rules of Sanskrit morphology

• **First practical Model** - Koskenniemi’s two-level morphology
Approaches for MA

• Koskenniemi’s two level morphology
  – PC-KIMMO
  – ENGTWOL

• Panini grammar framework
  – Anusaaraka Morphological analyzer

• Affix spiriting approach
  – Arabic Morphological Analyzer
Syntax Analysis

• Analysis structure in the text and is used to determine whether or not a text conforms to an expected format

• Parsing Techniques
  – Top-down parsers
  – Bottom-up parsers
Semantic Analysis

• Handling of meaning
• This is the most difficult aspect of all natural languages
• Most translation systems generally show issues at this level
Multi Agent System (MAS)

- Meaning of a sentence is a result of lexical, syntax and semantics definition of a language

- Therefore Language primitives can be considered as agents and meaning is a result of their collaboration

- Thus the meaning of a sentence can be handled through the communication among such agents
Features of the MAS

• Communication
• Autonomy
• Adaptability
• Reactivity
• Proactivity
Overview of the English and Sinhala Languages
English Language

• Eight parts of speech
  – Noun, Pronoun, Verb, Adjective, Adverb, Preposition, conjunction and interjection

• English Grammar has Minimal inflection

• English Morphology is simpler than the other Indo-European languages

• Morphological Components
  – Stems
  – Affixes (Prefixes, suffixes, infixes and circumfixes)
  – Example
    Rewrites (Re-Prefix, write – Stem, s – suffix)
English Language Syntax

• English uses SVO word order
• English language consist with 12 tenses
• Types of English sentences
  – Simple Sentence
  – Compound Sentence
  – Complex Sentence
• Simple Sentence
  – Declarative Sentence
  – Interrogative Sentence
  – Imperative Sentence
  – Conditional Sentence
Sinhala Language

- Sinhala language has its own writing system, which is an offspring of the Brahmi script.
- Sinhala alphabet consists of 61 letters comprising 18 vowels, 41 consonants and 2 semi-consonants.
- It contains a pair of vowel sounds that are unique to it.
  - Short vowel: ‘ɐ’ (ae)
  - Long vowel: ‘ɐː’ (aae)
- Sinhala is an inflection rich language.
- Participates inflection, derivation and conjugation for nouns and verbs.
Morphological Components

• **Prakurthi** (Prakurthi)
  - को (Nama)
  - क्रिया (Kriya)
  - गुण (Guna)
  - विलास (Vilasa)
  - निपथ (Nipatha)

• **Prathya** (Prathya)
  - नाम (Nama)
  - विभक्ति (Vibakthi)

• **Upasarga** (Upasarga)

• **Thadhitha** (Thadhitha)
Sinhala Language Syntax

• Sinhala uses SOV word order
• Sinhala sentence contains 8 components,

1  අපා සිදුකළම
2  සිදුකළම
3  පැර සිදුකළම
4  සිදුකළම
5  දෙරෙහෙ සිදුකළම
6  දෙරකු අරම්ව සිදුකළම
7  දෙරකු නීරාන
8  දෙරකු සේසු
Comparisons between English and Sinhala

• **Fundamental Differences**
  – English contains 8 parts of speech and Sinhala contains only 4 parts of speech
  – Sinhala has five nasal sounds ے (nng), මැ (ndj), ව (nnd), ත (nd), ඔ (mb).
Comparisons between English and Sinhala (Contd..)

- **Morphological Differences**
  - The English uses Stems Affixes to generate the English words
  - The Sinhala uses Prakurthi, Prathya, Upasarga and Thadhitha to generate Sinhala words
  - “Sandhi rules” are used to join two or more words
Comparisons between English and Sinhala (Contd..)

• Syntax in two languages
  – English language uses SVO word order and Sinhala language uses SOV word order
  – Sinhala language has only 3 tenses where English language uses 12 tense forms
  – Structure of a sentence contains fundamental differences in both two languages
Approach
Novel Approach

Theoretical basis of the novel approach comes from the concept of Varanageema (conjugation) in Sinhala Language.
Concept of Varanegeema

• Varanegeema (conjugation) in Sinhala language presents how we can drive various word forms from a given base word
  – 27 forms for a Noun
  – 36 forms for a Verb

• Conjugation handles Person, Preposition, Tense, Singular/Plural, and Active/Passive
<table>
<thead>
<tr>
<th>හේතු</th>
<th>හේතුං</th>
<th>හේතුහාදී</th>
<th>හේතුහාදී</th>
<th>හේතුහාදී</th>
</tr>
</thead>
<tbody>
<tr>
<td>නිම්නාය</td>
<td>නිම්න</td>
<td>නිම්නාය</td>
<td>නිම්නාය</td>
<td>නිම්නාය</td>
</tr>
<tr>
<td>මන්දිරා</td>
<td>මන්දිරා</td>
<td>මන්දිරා</td>
<td>මන්දිරා</td>
<td>මන්දිරා</td>
</tr>
<tr>
<td>මැතිවලි</td>
<td>මැතිවලියේ</td>
<td>මැතිවලියේ</td>
<td>මැතිවලියේ</td>
<td>මැතිවලියේ</td>
</tr>
<tr>
<td>මහානායකයේ</td>
<td>මහානායකයේ</td>
<td>මහානායකයේ</td>
<td>මහානායකයේ</td>
<td>මහානායකයේ</td>
</tr>
<tr>
<td>මිදුලු</td>
<td>මිදුලු</td>
<td>මිදුලු</td>
<td>මිදුලු</td>
<td>මිදුලු</td>
</tr>
<tr>
<td>කුලී</td>
<td>කුලීය</td>
<td>කුලීය</td>
<td>කුලීය</td>
<td>කුලීය</td>
</tr>
<tr>
<td>යුතුමා</td>
<td>යුතුමා</td>
<td>යුතුමා</td>
<td>යුතුමා</td>
<td>යුතුමා</td>
</tr>
<tr>
<td>වෙනිහි</td>
<td>වෙනිහි</td>
<td>වෙනිහි</td>
<td>වෙනිහි</td>
<td>වෙනිහි</td>
</tr>
<tr>
<td>උස්සාදී</td>
<td>උස්සාදී</td>
<td>උස්සාදී</td>
<td>උස්සාදී</td>
<td>උස්සාදී</td>
</tr>
<tr>
<td>මහාපාදයේ</td>
<td>මහාපාදයේ</td>
<td>මහාපාදයේ</td>
<td>මහාපාදයේ</td>
<td>මහාපාදයේ</td>
</tr>
<tr>
<td>දකුණු</td>
<td>දකුණු</td>
<td>දකුණු</td>
<td>දකුණු</td>
<td>දකුණු</td>
</tr>
<tr>
<td>මහාස්කෘතියේ</td>
<td>මහාස්කෘතියේ</td>
<td>මහාස්කෘතියේ</td>
<td>මහාස්කෘතියේ</td>
<td>මහාස්කෘතියේ</td>
</tr>
<tr>
<td>මහාදේශයේ</td>
<td>මහාදේශයේ</td>
<td>මහාදේශයේ</td>
<td>මහාදේශයේ</td>
<td>මහාදේශයේ</td>
</tr>
<tr>
<td>ප්‍රධාන</td>
<td>ප්‍රධානා</td>
<td>ප්‍රධානා</td>
<td>ප්‍රධානා</td>
<td>ප්‍රධානා</td>
</tr>
<tr>
<td>සමුදාය</td>
<td>සමුදායේ</td>
<td>සමුදායේ</td>
<td>සමුදායේ</td>
<td>සමුදායේ</td>
</tr>
<tr>
<td>මහාඋස්ථානයේ</td>
<td>මහාඋස්ථානයේ</td>
<td>මහාඋස්ථානයේ</td>
<td>මහාඋස්ථානයේ</td>
<td>මහාඋස්ථානයේ</td>
</tr>
<tr>
<td>විහාරයේ</td>
<td>විහාරයේ</td>
<td>විහාරයේ</td>
<td>විහාරයේ</td>
<td>විහාරයේ</td>
</tr>
<tr>
<td>විශේෂ</td>
<td>විශේෂා</td>
<td>විශේෂා</td>
<td>විශේෂා</td>
<td>විශේෂා</td>
</tr>
<tr>
<td>මාතරු</td>
<td>මාතරුව</td>
<td>මාතරුව</td>
<td>මාතරුව</td>
<td>මාතරුව</td>
</tr>
</tbody>
</table>
Sinhala Verb Conjugation

<table>
<thead>
<tr>
<th>කිං</th>
<th>අංග</th>
<th>හෝ.</th>
<th>අංග</th>
</tr>
</thead>
<tbody>
<tr>
<td>ගැටත්</td>
<td>ගැටත්</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ගේ</td>
<td>ගේ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ගෙ</td>
<td>ගෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ව මෙ</td>
<td>ව මෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ව මෙ</td>
<td>ව මෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ව මෙ</td>
<td>ව මෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ගැටත්</td>
<td>ගැටත්</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ගේ</td>
<td>ගේ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ගෙ</td>
<td>ගෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ව මෙ</td>
<td>ව මෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ව මෙ</td>
<td>ව මෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
<tr>
<td>ව මෙ</td>
<td>ව මෙ</td>
<td>අංග</td>
<td>අංග</td>
</tr>
</tbody>
</table>
### Nama Prakurthi Gana (Nama ආණෝ චාක්කි මාද)

<table>
<thead>
<tr>
<th>ගාණය</th>
<th>අංකයේ මියක මෝජ</th>
<th>අරුණාක්ෂය මෝජ</th>
<th>වේදීමේ මෝජ</th>
<th>ගාණය</th>
</tr>
</thead>
<tbody>
<tr>
<td>ගේ පු දරුම</td>
<td>ගේ පු දරුම</td>
<td>ගේ පු දරුම</td>
<td>ගේ පු දරුම</td>
<td>ගේ පු දරුම</td>
</tr>
<tr>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
</tr>
<tr>
<td>මාදු ඇති</td>
<td>මාදු ඇති</td>
<td>මාදු ඇති</td>
<td>මාදු ඇති</td>
<td>මාදු ඇති</td>
</tr>
<tr>
<td>අංකයේ මෝජ</td>
<td>අංකයේ මෝජ</td>
<td>අංකයේ මෝජ</td>
<td>අංකයේ මෝජ</td>
<td>කුඹ්බා දරුම</td>
</tr>
<tr>
<td>මාදු ඇති</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
</tr>
<tr>
<td>මාදු ඇති</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
</tr>
<tr>
<td>මාදු ඇති</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
</tr>
<tr>
<td>මාදු ඇති</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
</tr>
<tr>
<td>මාදු ඇති</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
<td>කුඹ්බා දරුම</td>
</tr>
</tbody>
</table>
## Nama Prakurthi Gana ( nama නැමෙහි නාමය)

<table>
<thead>
<tr>
<th>Base Form</th>
<th>Add</th>
<th>Remove</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>විශේෂ වැඹෙ</td>
<td>ඝ</td>
<td>ප</td>
<td>අශ්කාබුෂාන්තිය</td>
</tr>
<tr>
<td>අහියෝ ආවේ</td>
<td>ඝවස් ඝ</td>
<td></td>
<td>අහියෝ ආවේන්කිය</td>
</tr>
<tr>
<td>රූයෝ විදේ</td>
<td>ඝවස් ඝ</td>
<td></td>
<td>රූයෝ විදේවිය</td>
</tr>
<tr>
<td>මුළු ආවේ</td>
<td>ඝවස් ඝ</td>
<td></td>
<td>මුළු ආවේවිය</td>
</tr>
<tr>
<td>මුළු වෙදිනා</td>
<td>ඝ්වෙති ඝ</td>
<td></td>
<td>මුළු වෙදිනාවිය</td>
</tr>
</tbody>
</table>
BEES

-Bilingual Expert for English to Sinhala-
BEES

• **Input**
  – Grammatically correct English sentence(s)
  – HTML document with English text
  – Selected English text from any source

• **Output**
  – Normal Sinhala Wijesekara key layout
  – HTML format (With normal Sinhala fonts)
  – Sinhala Unicode
Features

- BEES is primarily a rule-based System
- BEES also uses context based and human-assisted strategies for translations.
- BEES is built on the concept of Varanageema
- BEES has been implemented as a Multi Agent System
Features (cont...) 

• Due to Varanageema concept, Sinhala Lexical resources consume very little memory space.
• It also uses minimal computer resources as the agents initialize only when necessary, execute and terminate at the end.
• BEES can work proactively and reactively
Features (cont…)

• BEES can be used as an stand alone application as well as web-based application
• BEES can be used as a translation plugin for any text processing application
• BEES has been implemented to run on both Windows and Linux
• BEES is a Prolog based system with Java support
• BEES provides built-in tools for maintenance, evaluation and updating of the system
Design and Implementation
Top Level Design

Input English Sentence(s)

English Language system
(Morphological analysis & Syntax analysis)

English to Sinhala Translation system
(Base word translation)

Sinhala Language system
(Morphological generation & Syntax generation)

Output Sinhala Sentence(s)

English Dictionary

Bilingual and concept dictionaries

Sinhala Dictionary
Components

9 Agents
1. English Morphological Analyzer Agent
2. English Parser Agent
3. English to Sinhala Based Word Translator Agent
4. Sinhala Word Generator Agent
5. Sinhala Parser Agent
6. Transliteration Agent
7. Intermediate Editor Agent
8. Message Space Agent
9. Request Agent

3 Supporting Agents
1. Dictionary Updating Agent
2. Sinhala Word Generator Agent
3. Online Search Agent

4 Dictionaries (Ontology)
1. English Dictionary
2. Sinhala Dictionary
3. English- Sinhala Bilingual Dictionary
4. Concept Dictionary

6 Prolog based Modules
1. English Morphological Analyzer
2. English Parser
3. Base word Translator
4. Sinhala Morphological Generator
5. Sinhala Parser
6. Transliteration module
Agents
English Morphological Analyzer Agent

- Automatically creates for each word in the input sentence
- Access **English Morphological analyzer** and reads the result of the morphological analysis
- After analyzing the English word, agents are self destroyed from the system
- Implemented by using Java
English Parser Agent

• Analyzes the input sentence by using English parser

• English Parser agent accesses the English parser and reads the result of the Syntax analysis

• After analyzing the English sentence, Self destroyed from the system

• Implemented by using Java
English to Sinhala Base Word Translator Agent

- Automatically creates for a given English word

- Uses **base-word translator** to translate English base-word into Sinhala base-word

- Uses Bilingual dictionary and concept dictionary for proper translation

- Implemented using JAVA
Sinhala Morphological Generator Agent

- Creates for each Sinhala base-word which is provided by the base word translator

- Access the **Sinhala Morphological generator** and reads the output result of the Sinhala Morphological generation

- Implemented using JAVA
Sinhala Parser Agent

- Accesses the **Sinhala parser** and generates grammatical correct Sinhala sentence
- After Generating the Sinhala sentence, Self destroyed from the system
- Implemented using JAVA
Transliteration agent

- Transliterate English text into Sinhala by using **Transliteration module**

- Transliteration is used as a solution for the Out-of-vocabulary problem

- After Generating the Sinhala sentence, Self destroyed from the system
Intermediate Editor Agent

- Facilities to edit the text by using human support through the **intermediate editor**
- Showing synonyms, related words
- Implemented Using JAVA
Message Space Agent

- Enables communication among other agents
- Send messages for each agents
- Read the other agent state
- Control the translation process
- Active for a given request
- Implemented using JAVA
Request Agent

- Make a request and get the output results from other agents
- Implemented using JAVA
Supporting Agents

- Dictionary Updating Agent
- Sinhala Word Generator Agent
- Online Search Agent
Agent Implementation

- Implemented using JAVA thread technology
- 5 Public variables are used to control
  - Message
  - State
  - Live
  - waitTime
  - Input
- 3 constructors have been used to initialize
- Updates all the process and running state on the xml file
## Agent States and Its Meaning

<table>
<thead>
<tr>
<th>Agent State</th>
<th>Meaning of the state</th>
</tr>
</thead>
<tbody>
<tr>
<td>dead</td>
<td>Agent is dead</td>
</tr>
<tr>
<td>ready</td>
<td>Agent is ready to work</td>
</tr>
<tr>
<td>busy</td>
<td>Agent is busy (Cannot access in this time)</td>
</tr>
<tr>
<td>sdead</td>
<td>Agent dead after successfully completing the task</td>
</tr>
<tr>
<td>edead</td>
<td>Agent dead with some error of the given task</td>
</tr>
<tr>
<td>error</td>
<td>Agent in the error state</td>
</tr>
<tr>
<td>running</td>
<td>Agent is running</td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<agents>
  <ema msg="Sucss" name="ema" state="sdead" words="boy eats rice. It is good"/>
</agents>
```
class EnglishMorpion extends Thread {
    String message;           // Message of the agent
    String state;             // State of the agent
    boolean live = true;      // State of the agent
    int waittime = 250;       // Delay time
    String words = "";         // Input Sentence
    public void run()        // Thread options
    public synchronized void AnalyzeEnglishWords(String WordList)
    EnglishMorpion(String wd);
    EnglishMorpion();
    EnglishMorpion(String mg, String st);
    static public void main(String args[]);
}
Prolog Based Modules

1. English Morphological Analyzer
2. English Parser
3. Base word translator
4. Sinhala Morphological Generator
5. Sinhala Parser
6. Transliteration Module
English Morphological Analyzer

- Reads a given English sentence word by word and identifies morphological information on each word.
- Customized an existing English morphological analyzer.
- Implemented using SWI-Prolog.

**Input English word:** Boy

<table>
<thead>
<tr>
<th>Category</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Noun</td>
</tr>
<tr>
<td>ID</td>
<td>E1000006</td>
</tr>
<tr>
<td>Person</td>
<td>3rd</td>
</tr>
<tr>
<td>Number</td>
<td>Singular</td>
</tr>
<tr>
<td>Sex</td>
<td>Male</td>
</tr>
<tr>
<td>Case</td>
<td>Dative</td>
</tr>
<tr>
<td>Word</td>
<td>Boy</td>
</tr>
</tbody>
</table>

eng_noun([e1000006], td, sg, ma, sb, 'boy').
English Parser

- Functions as a syntax analyzer
- Receives source English sentences and tokens from English Morphological Analyzer
- Customized an existing parser
- Implemented using SWI-PROLOG

Sentence: A boy eats red rice

```
eng_sentence_type(simple, sp).
eng_sen_verb([e1000009]).
eng_sen_complement([e1000008, e1000013]).
eng_sen_subject([e1000001, e1000006]).
```

Diagram: Syntax tree with labeled nodes for sentence construction.
Base-word Translator

- Translate English base word into Sinhala base word with the help of bilingual and concepts dictionaries.
- Use word to word translation at this level.

Sentences:

- A boy eats red rice
  
  + eng_detm([e1000001], id, 'a').
  eng_noun([e1000006], td, sg, ma, sb, 'boy').
  eng_verb([e1000009], sp, 'eats').
  eng_adjv([e1000008], p, 'red').
  eng_noun([e1000013], td, sg, no, sb, 'rice').

- Bilingual Dictionary
- Concepts Dictionary

estrwords(1001, e1000001, s1000000, dt).
estrwords(1002, e1000006, s1000014, na).
estrwords(1003, e1000009, s1000011, vb).
estrwords(1004, e1000008, s1000029, aj).
estrwords(1005, e1000013, s1000015, na).
Sinhala Morphological Generator

- Uses concepts of Varanegeema
- Reads the words from translator word by word
- Generates appropriate Sinhala terms according to grammatical information
- Implemented using SWI-Prolog.

English word: Boy
eng_noun([e1000006], td, sg, ma, sb, 'boy').
+ eng_sen_verb([e1000009]).
  eng_sen_complement([e1000020]).
  eng_sen_subject([e1000001, e1000006]).
+ estrwords(1002, e1000006, s1000014).

Sinhala Dictionary

Category | Data
---------|------
Type | Noun
ID | S1000014
Person | 3rd
Number | Singular
Sex | Male
Case | Subject (Katru)-v1
Option | හුරිහුණුව
Word | විශේෂ
Sinhala Parser

Sinhala Morphological Info+

Syntax info

Sinhala sentence

PROLOG

sinsentencewd(['ходит, имеет', 'ах', 'есть', 'верить', []]).
Transliteration Module

- Developed two types of transliteration models.
  - Original English text into Sinhala transliteration
  - Sinhala words written in English characters transliterate into Sinhala characters.

- Finite State Transducers (FST) are used to develop these two modules
- Implemented using SWI-PROLOG
Lexical Dictionaries (Ontology)

1. English Dictionary
2. Sinhala Dictionary
3. English-Sinhala Bilingual Dictionary
4. Concept Dictionary
English Dictionary

• Designed as a prolog database
• Based on structure of the standard WorldNet
• Regular Nouns, Regular Verbs, Irregular nouns and irregular verbs are stored separately
• Implemented using SWI-PROLOG
English Dictionary

• Implemented through the 5 prolog data files
  – eng_irr_noun.pl’, eng_reg_noun.pl, eng_irr_verb.pl,
    eng_reg_verb.pl and eng_irr_word.pl.
• Irregular noun information (eiw/7 )
  – eiw(4000001, na, fs, sg, co, sb, 'i').
• English regular nouns
  – erw(1000001, na, ma, 'boy').
• English irregular verb.
  – eiw(5000001, vb, if, 'eat').
• English regular verb
  – erw(2000001, vb, 'play').
• Other words
  – eiw(3000004, aj, p, 'good').
Sinhala Dictionary

- Designed as a prolog database
- Regular Nouns, Regular Verbs, Irregular nouns and Irregular verbs are stored separately
- Rules are also included
  - Nama Gana
  - Kriya Gana
  - Vibakthis forms
- Implemented using SWI-Prolog
Sinhala Dictionary

- Implemented through the 6 prolog data files
  sin_reg_nouns.pl, sin_irr_nouns.pl, sin_reg_verb.pl, sin_irr_verb.pl,
  sin_irr_words.pl, sin_case_rules.pl and sin_rule_dic.pl

- Predicate samples
  - sn([1000001], td, ma, li, s900004, s910000, s910000, s910000, 'හිසිමා').
  - sn([4000001], fs, sg, co, li, dr, v1, 'ය').
  - sfv([5000001], s910001, s910002, s910001, s910001, s910001, s910001, s910001, s910001, 'කොල්').
  - sfv([8000002], fs, sg, at, pr, 'කොල්').
  - noun_posfix(s930003, li, bas, 'ා', 'ව').
  - noun_posfix(s930003, li, sds, 'ා', 'ව').
  - noun_posfix(s930003, li, sdo, 'ා', 'ව').
  - noun_posfix(s930003, li, sis, 'ව/ෝ', 'ව').
  - noun_posfix(s930003, li, sio, 'ව/ෝ', 'ව').
  - noun_posfix(s930003, li, pdo, 'ළමයා', 'ව').
  - noun_posfix(s930003, li, pdo, 'ළමයා', 'ව').
English Sinhala Bilingual Dictionary

• Designed as a prolog database
• Used to identify appropriate Sinhala base word for the given English word
• Shows relations between English and Sinhala words
• Implemented using SWI-Prolog
• Sample codes
  – esw(6000006, 1000001, 1000001, na, 'boy', 'හොඳය').
  – esw(6000004, 5000001, 5000001, vb, 'eat', 'කනව').
  – esw(6000006, 3000009, 3000009, aj, 'fat', 'මහත').
  – esw(6000006, 3000009, 3000009, av, 'daily', 'නපතාම').
Concept Dictionary

- Designed as a prolog database
- Is used to identify the semantics of the words
- Context information for the Sinhala words
- Sample code
  - `eng_cons_word(e1000000, s1000000, e1000000).`
Evaluation
Evaluation

- Conducted a white box testing approach and tested each module in the machine translation system through the developed testing tools (Module testing)

- Evaluated the system performance and calculated the error rate through the evaluation test bed (Performance testing)

- Intelligibility and accuracy test was conducted through the human support (Accuracy testing)
Module Testing

• Test each Module through the developed testing tools
  1. English Morphological analyzer
  2. English parser
  3. Base word Translator
  4. Sinhala Word Generator
  5. Sinhala Parser
  6. Transliteration modules
Performance Testing

• The evaluation test bed has been implemented as an experimental setup
• Following questions to evaluate the translation.
  – Subject Verb Agreement (correct/incorrect)
  – Tense of the Sentence (correct/incorrect)
  – Word Conjugation (all correct/some are correct/all incorrect)
  – Word Order in the Sentence (correct/incorrect)
  – Meaning of the translated sentence
    • 0 – Error
    • 1 - Meaningless
    • 2 - Basically OK
    • 3 - Perfect
Error Rates

**Word Error Rate**

\[
\text{Word Error Rate} = \frac{\sum \text{Number of incorrect words in the sentence}}{\sum \text{Number of words in the sentence}}
\]

**Sentence Error Rate**

\[
\text{Sentence Error Rate} = \frac{\sum \text{Number of incorrect sentences}}{\sum \text{Total Number of Sentences}}
\]
Accuracy Testing (contd..)

- 200 sample sentences are collected and grouped into 20 sets (10 sentences for each group)
- Each sentence is translated using BEES
- Each set of sentences is given to the human translator and scored for each sentence
  - Subject Verb Agreement (correct/incorrect)
  - Tense of the Sentence (correct/incorrect)
  - Word Conjugation (all correct/some are correct/all incorrect)
  - Word Order in the Sentence (correct/incorrect)
  - Meaning of the translated sentence
  - 0 - Error
  - 1 - Meaningless
  - 2 - Basically OK
  - 3 - Perfect
<table>
<thead>
<tr>
<th></th>
<th>Past Tense</th>
<th>I wrote a new book</th>
<th>Past Perfect</th>
<th>I had written a book</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td>ඔමු නොලොම් වීම කියන්නේ</td>
<td>Past Perfect Continuous</td>
<td>I had been writing a book</td>
</tr>
<tr>
<td>6</td>
<td>Past Continuous</td>
<td>I was writing a new book</td>
<td></td>
<td>I will write a new book</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ඔමු නොලොම් ලකුණ කියන්නේ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation Results - Module Testing

- Sinhala transliteration
- Sinhala sentence generation
- Sinhala Morphological generation
- English to Sinhala base-word translation
- English Syntax analysis
- English Morphological Analysis

Percentage (%)
### Evaluation Results – Performance Testing

<table>
<thead>
<tr>
<th>Case</th>
<th>Results</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct subject verb agreement</td>
<td>186</td>
<td>93</td>
</tr>
<tr>
<td>Correct tense translation</td>
<td>190</td>
<td>95</td>
</tr>
<tr>
<td>Correct Noun verb generation</td>
<td>180</td>
<td>90</td>
</tr>
<tr>
<td>Correct word order</td>
<td>185</td>
<td>92.5</td>
</tr>
<tr>
<td>Total number of sentences</td>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>
# Evaluation Results - Accuracy Testing

<table>
<thead>
<tr>
<th>Test case</th>
<th>Sentences</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect translation</td>
<td>143</td>
<td>71</td>
</tr>
<tr>
<td>Basically OK</td>
<td>52</td>
<td>26</td>
</tr>
<tr>
<td>Meaningless</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Error</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

**Translation Accuracy**

- Perfect translation: 71%
- Basically OK: 26%
- Meaningless: 2%
- Error: 3%

![Translation Accuracy Pie Chart](chart.png)
Final Results

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Error Rate (WER)</td>
<td>7.2 %</td>
</tr>
<tr>
<td>Sentence Error Rate (SER)</td>
<td>5.4 %</td>
</tr>
<tr>
<td>Intelligibility and Accuracy</td>
<td>89.1 %</td>
</tr>
</tbody>
</table>
### Evaluator’s Comments

<table>
<thead>
<tr>
<th></th>
<th>Present continuous</th>
<th>The beautiful girl is singing a song</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>සතාවක් ගොඩනැගීමෙන් ක්‍රමාංකයක් නිර්මාණයක් සිදු කරන්නේ සිතියි.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Present perfect</th>
<th>We have written new books</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td>මගේ කාර්ය ලියා කියුණේ අනුබෝධයක්</td>
</tr>
</tbody>
</table>

- **Error** ✔️ Meaning less ☐ Basically OK ✔️ Perfect ☐

<table>
<thead>
<tr>
<th></th>
<th>Passive future perfect</th>
<th>A book shall have been written by me</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>කොළඹ දැක්වී සැකසී මගේ ගොඩනැගීමෙන් නිර්මාණයක්</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>grammar</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 2 3 4</td>
</tr>
</tbody>
</table>

**Suggestions:**
- Follow the same sentence in different tenses. This will help to understand the differences than this.
Conclusion
Conclusion

Achieving Objectives

Objective 1: Critically review the existing systems for machine translation.

Critically review the existing translation approaches and 35 successful systems have been discussed (chapter 2)
Conclusion (contd..)

Achieving Objectives

Objective2:

Study the concepts / techniques for Natural Language Processing

Critically reviewed the existing concepts/techniques for Natural language processing with more attention on the machine translation (Chapter 2)
Conclusion (Cont..)

Achieving Objectives

Objective3:

Study the concepts/techniques and adapt existing Morphological Analyzers and Parsers for English language.

There are number of technologies available to develop morphological analyzers and parsers for machine translation. Critically reviewed the existing concepts/techniques (Chapter 2).
Conclusion (Cont..)

Achieving Objectives

Objective4:
  Design and develop Morphological Analyzer/Generator and Parser/Composer for Sinhala Language

  With the theoretical basics of the Sinhala language, Sinhala morphological generator and the Sinhala sentence composer have been developed
Conclusion (Cont.)

Achieving Objectives

Objective5:
Design and develop lexical databases for English to Sinhala Machine Translation

Designed and developed four lexical dictionaries namely English dictionary, Sinhala dictionary, English to Sinhala bilingual dictionary and the concept dictionary
Conclusion (Cont..)

Achieving objectives

Objective 6:

Design and develop English to Sinhala Machine Translation system by integrating above sub-systems to form the Multi agent system

The English to Sinhala machine translation system has been designed and developed as a multi agent system
Conclusion (Cont..)

Achieving Objectives

Objective 7:
Evaluate the system

The English to Sinhala machine translation system has been evaluated through the three stages

– Module testing
– Performance testing
– Accuracy testing)
Overall Conclusion

- System shows 89.1 accuracy with 7.2% word error rate and the 5.4% sentence error rate
- The system successfully translates the English sentence through the concept of Varanegeema
Limitations

• Translation system works on the simple sentences
• System does not handle multi-word expressions, idioms and compound sentences
• The current lexical resources in the system are limited
Further works

- Concept of Varanegeema (conjugation) can be tried out for machine translation systems that deal with languages closer to Sinhala
- Can be expanded with more lexical resources
- Handling compound sentences
- Expansion to the parser for handling more grammatical structures
- Sinhala to English machine translation
References


References (contd..)


References (contd..)


References (contd..)


Demonstration
Thank You!

Thank You!