Sinhala Ontology Generator for English to Sinhala Machine Translation

B Hettige[1], AS Karunananda[2], G Rzevski[3]

[123] *Department of Computational Mathematics, University of Moratuwa, Sri Lanka*
[1] budditha@dscs.sjp.ac.lk, [2] asoka@itfac.mrt.ac.lk, [3] rzevski@gmail.com

Abstract*—Generating Ontology for a given word is a research challenging task that requires morphology, syntax, semantics and other linguistic knowledge. This paper presents a multi-agent system, named Sinhala Ontology Generator which is capable to generate ontology for a given Sinhala word. Ontology of the Sinhala word has been categorized by using 29 Sinhala part of speech tags including Noun, Verb, Adjective and Adverb. To generate Ontology for the given Sinhala word(s) Morphological, Grammatical, Semantic and Special attributes have been considered. The Ontology generator has been designed through the java based multi-agent system development framework named MaSMT. Sinhala Ontological generator communicates with Sinhala Morphological analyser, Sinhala syntax analyser, dictionary reader and word reader to collect information to generate the Ontology. The system has been tested with 400 sample Sinhala words. Experimental result indicates that morphologically and syntactically identified words give more than 70% of accuracy and unseen words give less than 50% of accuracy.*

Keywords— Machine Translation, Multi-agent Systems, Sinhala Language

I. INTRODUCTION

Ontology represents knowledge as a hierarchy of concepts. These Ontologies can be used to describe grammatical, morphological, syntactic, semantic and other special features of a word. Hence, ontology of a word (word ontology) is a valuable and essential resource to develop Natural Language Processing (NLP) applications such as machine translation systems, natural language understanding systems, word segmentation systems, parsing systems etc.

At present, many researchers are researching to develop NLP applications through the ontology based language modelling. There are number of lexical dictionaries and WorldNet have been developed through the language ontology. In the Indian region, Word-Hindi dictionary has been developed by the Centre for Indian Language Technology IIT Bombay, for the purpose of machine translation (CFILT 2012). Further, number of knowledge-based machine translation systems including PAN-GLOSS are used this ontology concept to model its language knowledge (Knight and Steve 1994).

In Sri Lanka numbers of Sinhala language resources are available to develop NLP applications including Sinhala corpus and lexicon (LTRL 2011)., few English/Sinhala dictionaries (Madhura 2010) , Sinhala part of speech tagger(Jayasuriya and Weerasinghe 2013), Sinhala WorldNet (WordNet 2013), Rule based English to Sinhala Machine Translation system (Hettige and Karunananda 2011) etc. However, ontology model of the Sinhala words for the purpose of machine translation is still not developed.

This paper presents a multi-agent system, named Sinhala ontology generator (SOG) which is capable to generate ontology for a given Sinhala word. Ontology of a Sinhala word has been categorized using 29 tags which is based on the existing tagging system; developed by the UCSC Sinhala part of speech tagger (Jayasuriya and Weerasinghe 2013). These tags are used to identify Sinhala Noun, Verb, Adjective, Adverb etc. To generate Sinhala word ontology Morphological, Grammatical, Semantic and Special attributes have been considered. The Sinhala Ontology generator has been designed through the MaSMT framework (Hettige et al. 2013). This ontology generator has an ability to communicate with other MaSMT systems such as Morphological analyser (Hettige et al. 2012), Sinhala syntax analyser, Sinhala dictionary reader and the Sinhala word reader. The Morphological analyser provides morphological information for the given Sinhala word. The Sinhala syntax analyser provides grammatical information for the given word(s). Sinhala Ontology generator uses Sinhala word reader to search usage of the given Sinhala word through the web. The Sinhala dictionary reader searches synonyms and antonyms word from online dictionaries. The Sinhala Ontology generator combines information provided by the above 4 systems (Morphological analyser, syntax analyser, Sinhala word reader and the Sinhala dictionary reader) and generates the relevant ontology.

The rest of the paper is organized as follows. Section 2 presents existing ontology modelling systems for Natural Language Processing. Section 3 gives grammatical review of the Sinhala language to model Sinhala Word Ontology. Section 4 gives design of the system including a brief description of each module. Then section 5 reports how system works for the given Sinhala word(s). Section 6 reports evaluation methodology for the SOG. Finally section 7 presents conclusions and further works of the research.

II. ONTOLOGY MODELLING SYSTEMS FOR NATURAL LANGUAGE PROCESSING

Natural language processing applications are used to analyse natural languages through the text or sound. Most of the natural language processing applications require complete language specific knowledge for the language analysis. Example: Machine translation system requires morphological, syntax and semantic knowledge to provide accurate translation. Ontology modelling is the successful way to model this language knowledge.

Worldwide, Large number of lexical dictionaries and wordNet resources have been developed using Ontology based language modelling concepts. Among others, word-hindi dictionary has been developed by center for Indian Language Technology (CFILT) IIT Bombay, for the purpose of machine translation. This word-Hindi dictionary has been design through the Noun, Verb, Adjective and Adverb Ontologies for the Hindi words by considering Morphological, grammatical and semantic categories.

In addition to the above word ontologies are widely used by the knowledge-based machine translation systems to handle their language knowledge. PAN-GLOSS is the large scale knowledge-based machine translation system which is used ontology concept to model its language knowledge (Knight and Steve 1994).

At present numbers of Sinhala language resources are available to develop NLP applications. Jayasuriya and Weerasinghe have developed part of speech (POS) tagger for Sinhala language (Jayasuriya and Weerasinghe 2013). This tagger is able to handle lexical items with multiple POS tags as well as predicting POS tags of previously unseen words. A stochastic approach, Hidden Markov Model (HMM) with tri-gram probabilities was used as the training and tagging model. The tagger achieved an overall accuracy of 62%.

Considering the existing Sinhala language based NLP applications, ontology model of the Sinhala words for the purpose of machine translation is still not developed. To design an ontology model for the Sinhala words it is essential to get knowledge about morphology and syntax about Sinhala language. The next section briefly describes Morphology and syntax of the Sinhala language.

III. SINHALA GRAMMAR

Sinhala is constitutionally recognized as the official language of Sri Lanka. Sinhala language consists of own writing system, which is an offspring from the Brahmi scripts (Balagalle 1995 ; Wikipedia 2014). However, it differs from all other Indo-Aryan languages. Sinhala language has a pair of vowel sounds that are unique to it, such as short vowel: 'ae' and Long vowel: 'aae'. Further, Sinhala is an inflationary rich language. It participates inflection, derivation and conjugation for nouns and verbs. Other words, adjective, adverb, and prepositions do not participate any inflection or derivations.

A. *Morphology of the Sinhala Language*

Morphology is the identification, analysis, and description of the structure of a given language's morphemes and other linguistic units, such as root words, affixes etc. (Wikipedia 2014). The Sinhala Language consists of four parts of speech namely noun, verb, preposition and other words.

The Sinhala noun consists of four types of inflections such as Gender, Number, Person (Purusha) and Case. The Sinhala nouns can be divided into three categories, namely, simple, complex and compound. A Simple noun contains only a prakurthi (base form) while a complex noun contains prakurthi and suffix (Prattya) (Disanayake 2000). A compound noun contains two or more prakruthi. Prakurthi is a base form of a word and is also in the non-inflection form. In addition to the above Upasarga and Thadditha change the meaning of a noun. All of these prakurthi, nama prattya, vibhaakthi prattya, thaddhitha and upasarga are morphos in Sinhala.

Further, there are 27 forms of nouns that can be generated by inflecting a single root word (prakurthi). This inflecting is called 'Nama varanagilla' (Word conjugation) (Hettige and Karunananda 2011). The Sinhala noun contains more than hundred rules to conjugate a noun using a given base form (Prakurthi). In Sinhala there are 15 conjugation patterns identified for generating a Sinhala noun. These patterns are called 'Gana'(Dissanayake 2000).

Sinhala verbs are divided into two general classes, namely, transitive verb (sakarmaka) and intransitive (Akarmaka). These two verb categories are inflected for voice (karaka), mood (vidi), tense (kala), number (wachana) and person (purusha). Voice can be either active or passive. There are four types of moods, namely, indicative, optative, imperative and conditional (Karunathilaka 2004). Sinhala verb consists of only three tenses namely Past tense, Present tense and future tense. Further main verb in a sentence (Akkyathaya) participates three types of inflections namely person, number and sex. Morphological point of view, preposition and other words do not participate any inflection or derivation.

B. *Syntax of the Sinhala Language*

Syntax teaches how sentences are constructed in conformity to the rule of grammar (Gunasekera 1986). According to the Sinhala grammar, Sinhala sentences can be categorized into six types such as simple, complex, contracted, collateral, compound and elliptical. The simple sentence contains only one subject and one finite verb. The complex sentence contains a principal sentence with one or more dependent or subordinate clauses. Subordinate clause can be divided into three parts such as Substantive clauses, Adjective clauses and Adverbial clauses.

There are 36 syntax rules in the Sinhala language to generate grammatically correct Sinhala sentences. Most of these rules represent subject verb agreement of the Sinhala sentences (Karunathilaka 2004).

C. *Semantics of the Sinhala Language*

Identification of the semantics of the Sinhala word is the most important part of the ontology generation. Linguistically, the meaning of a word exists on number of levels such as grammatical meaning, phrasal meaning, contextual meaning, idiomatic meaning, restricted meaning and proverbial meaning.

The grammatical meaning refers to the grammatical categories of the word such as noun, verb, adjective etc. As a simple example the Sinhala word 'කොස්ස' (Kossa) has several Sinhala meanings such as part of the wound or broom.

The phrasal meaning analyses the term of the grammatical function in phrase. In Sinhala language, there are large numbers of phrasal meanings. Further, identification of the contextual meaning, Idiomatic meaning, restricted meaning and proverbial meaning is more complex and a difficult task.

IV. DESIGN OF THE SINHALA ONTOLOGY GENERATOR

The Sinhala Ontology generator has been designed through the Multi-agent development framework MaSMT (Hettige 2013). The MaSMT framework provides two types of agents, namely ordinary agents and manager agents. The manager agent consists of number of ordinary agents within its control. Further, manager agents can directly communicate with other manager agents and each and every ordinary agent in the swarm is assigned to a particular manager agent. An ordinary agent in a swarm can directly communicate only with the agents in its own swarm and its manager agent. The framework has been implemented by using JAVA. It consists of 5 modules namely ordinary agents, manager agents, global message space, local message space and ontology. Design diagram of the MaSMT framework is shown in Fig1.

Agent Manager is an agent (java thread) of the system that manages its client agents. According to the MaSMT architecture, each manager can fully control its client agents. Therefore, manager can create, remove or control its client agent(s). The Manager agent in the MaSMT creates all its clients automatically at the initialization stage. This agent accesses the rule-base (agents' ontology) and assigns each rule for a client agent. It means, manager agent creates an agent for a rule which is available in the rule-base. In addition, manager can directly access its client agents and send messages directly. The Manager agent reads input message from the global message queue and assigns relevant tasks for the client agents. These messages are sent by the other manager agents in the Multi-agent system. Further, manager agent can control the priority of the client agents and the stage of the clients. This facility removes the unnecessary workload from its client agents.

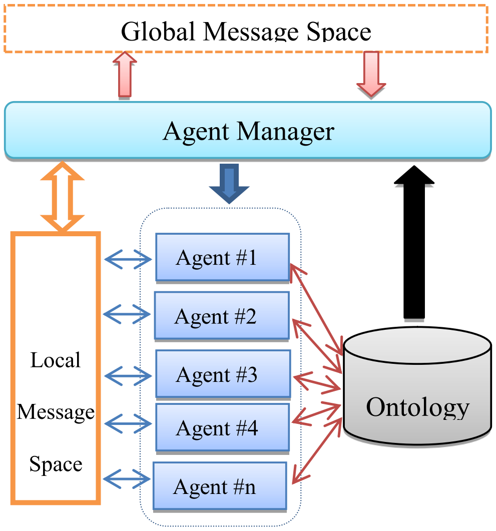
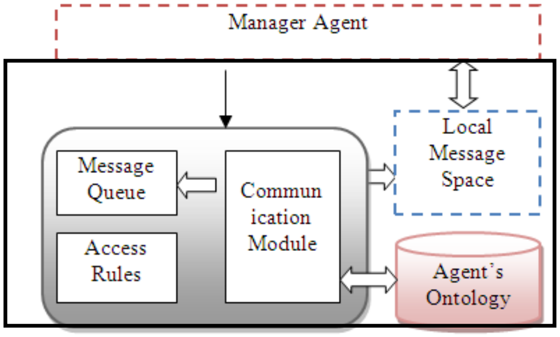


Figure 2. Design of the MaSMT Framework

The Ordinary agents are working under the control of the manager agent and each ordinary agent must have a manager agent. MaSMT agent is a simple java program (Thread) which supports limited task(s). These agents can communicate with each other through the messages space by using peer-peer or object-object communication methods. This ordinary agent consists of local message queue, access rule, communication module and the ontology. Fig. 2 shows the design of an ordinary agent.



These agents respond for the messages that are available in its local message queue. Each agent is assigned for only a limited task and it responses only for the assigned task (task is available as an access rule).

The communication module gives way to access ontology and the message spaces through a given media such as MySQL database Access, Objet-Object access, XML database access, peer-to-peer network access or client-server communication.

The Local message space is the visible area of each agent in the local agent group (Swam). Each agent can directly communicate with each other through the local message space. An agent in a given group has a local message queue with public access. Manager agent and other client agents in the group can directly access this message queue through the object-object communication method.

Global message space is used to communicate among managers in the MaSMT framework. This message space is visible only for the managers who are in the MaSMT framework. Managers send messages with the support of the communication module. In the distributed environment, managers work in different locations and communicate with each other through the client-server or peer-to-peer communication.

Ontology is the knowledge of each agent (ordinary and manager agents). Agent uses ontology to make the action. For instance, English morphological agents in the English to Sinhala machine translation system use English dictionary as the ontology. Morphological rules of Morphological agents are also stored in the ontology.

Messages are used to communicate among each other. These messages have been developed by using FIPA-ACL message standard. ACL Message consists with Participant in communication: sender, receiver, reply-to, Content of message: content, Description of Content: language, encoding, ontology. Control of conversation: protocol, conversation-id, reply-with, in-reply-to, reply-by etc. MaSMT uses MaSMTM message class to handle all the messages in the framework which is implemented based on ACL messages.

Communication modules are used to communicate among agents. These modules also give vital support to develop MaSMT as a distributed system. In addition to the common MySQL database access, Objet-Object access and XML database access, system can communicate with client-server or peer-to-peer network modes. Figure 3 shows the agent model of the Sinhala Ontology generator. Sinhala Ontology generator gets support from another 4 multi agent systems namely morphological analyser, syntax analyser, word reader and dictionary reader. Morphological analyser analyses given Sinhala word and provides morphological information. The syntax analyser tries to analyse given word list and provides grammatical information for the given word list (sentence or part of the sentence). The dictionary reader reads a given word form online dictionary and provides synonyms and anti-synonyms words. The word reader searches the word form online web resources and identifies usage of the Sinhala word. The Sinhala Ontology generator communicates with above 4 systems and generates the Sinhala word Ontology. Figure 4 shows the ontological category for a Sinhala noun.

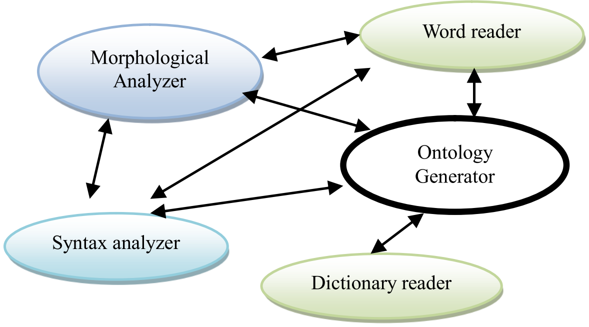


Figure 3. Multi-agent swarm

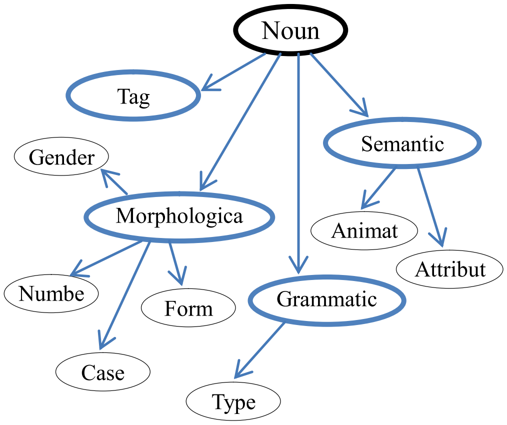


Figure 4. Ontological category for a Sinhala Noun

V. HOW SYSTEM WORKS

This section describes how system works for the given Sinhala words. SOG reads Sinhala word(s) as an input. Then SOG communicates with Sinhala Morphological analyser and collects morphological information. According to the morphological information SOG generates appropriates java object for Noun, verb, preposition, or other word. Then SOG communicates with Sinhala syntax generator, and identify grammatical category for the given Sinhala words. In this level SOG identify only phrase level syntax through the Sinhala syntax analyser. Then SOG completes the syntax of the ontology. After that SOG communicates with Sinhala word reader and dictionary reader to complete the relevant part of the ontology. Finally SOG generates the relevant XML file and Prolog (Swi-Prolog 2014) file according to the generated Ontology. The following sample shows some example for the generated XML file for the input Sinhala word 'මගේ පරිගණකය'(my computer).

```
<?xml version="1.0" encoding="UTF-8"?>
   <word        type="sinhala"        encoding="UTF-8"
fontset="unicode">
<data>මගේ</data>
  <tag>PRPC</tag>
  <morphology>
       <person>first</person>
       <number>single</person>
       <gender>conman</gender>
       <case>genitive </case>
       <form>adjective</from>
  </morphology>
       <syntax>
              <type>adjective</type>
```

```
                <base>පරිගණකය</base>
          </syntax>
          <semantic></semantic>
          <other></other>
     </word>

     <?xml version="1.0" encoding="UTF-8"?>
     <word        type="sinhala"        encoding="UTF-8"
fontset="unicode">
          <data>පරිගණකය</data>
          <tag>NNN </tag>
          <morphology>
                <person>third</person>
                <number>single</person>
                <gender>neuter</gender>
                <case>nominative </case>
                <form>root</from>
          </morphology>
          <syntax>
                <type>base</type>
          </syntax>
          <semantic></semantic>
          <other></other>
  </word>
```

VI. EVALUATION

Sinhala Ontology generator has been successfully tested by using 400 sample Sinhala words. In the first step, words are grouped into 16 sets (25 words for each set) and Sinhala word ontologies have been generated for the each word set (25 words of the each group are input for the SOG as one by one). These generated ontologies (including all categories of Sinhala words) have been manually tested by Sinhala subject expert.

In the second step, words are grouped in to 2, 3 and 4 words sets (meaningful sentence parts). Each word sets are given to SOG and results have been manually tested with the subject experts.

The First experimental result shows that morphologically and syntactically identified words give more than 70% of accuracy. However unseen words (unknown words) give less than 50% of accuracy. Further, second experimental result also shows that accuracy of the word ontology increases when more number of words are given at once.

VII. CONCLUSION AND FURTHER WORKS

This paper has reported a Sinhala Ontology Generator which can be used to generate ontology for the given Sinhala word(s). The Sinhala Ontology Generator has been designed as a multi–agent system through the MasMT framework. Sinhala Ontology Generator communicates with Morphological analyser, Syntax analyser, dictionary reader and word reader to generate appropriate Sinhala ontology. Ontology of a given Sinhala word has been developed by using morphological syntax and semantics information.

The SOG has been successfully tested with 400 sample Sinhala words. The Experimental result demonstrates that more than 70% accuracy for the morphologically and syntactically identified words and less than 50% accuracy for the unseen words (unknown words). In addition to the above, experimental results also show that accuracy of the ontology increases when more number of words are given at once.

This Sinhala Ontology generator can be used as a supporting tool for the English to Sinhala agent based machine translation system to generate agent’s ontology.

REFERENCES

Balagalle VG. (1995). Basha Adauanayasaha Sinhala Vivaharaya, S. Godage and Brothers, Colombo 10, Sri Lanka.

CFILT,(2012). <http://www.cfilt.iitb.ac.in/index.html> Home page, Center for Indian Language Technology, CFILT, IIT Bombay,2014. Accessed 28 May 2014.

Disanayake JB, (2000). BasakaMahima 6: Prakurthi, Colombo 10, Sri Lanka : S.Godage and Brothers.

Gunasekera AM. (1986). A Comprehensive Grammar of the Sinhalese Language, New Delhi, India AES Reprint.

Hettige B and Karunananda AS , “Computational model of grammar for English to Sinhala Machine Translation,” in 2011 International Conference on Advances in ICT for Emerging Regions (ICTer), 2011, pp. 26–31.

Hettige B and Karunananda AS. (2011). A Word as an Agent for Multi-agent based Machine Translation, Proceedings of the ITRU Research Symposium, Moratuwa, Sri Lanka.

Hettige B, Karunananda AS and Rzevski G, (2012). Multi-agent System Technology for English-Sinhala

Morphological Analysis Proceedings of the Ninth Annual Sessions, Sri Lanka Association for Artificial Intelligence (SLAAI).

Hettige B, Karunananda AS, Rzevski G, (2013), MaSMT: A Multi-agent System Development Framework for English - Sinhala Machine Translation, International Journal of Computational Linguistics and Natural Language Processing Vol 2 Issue 7 July 2013.

Jayasuriya M and Weerasinghe AR. (2013). Learning A Stochastic Part Of Speech Tagger For Sinhala, International Conference on Advances in ICT for Emerging Regions, ICTer2013 Colombo, Sri Lanka.

Karunathilaka WS. (2004), Sinhala Basha Viharanaya, Colombo, Sri Lanka: M. D. Gunasena and Ltd.

Knight K and Steve KL. (1994), building a large scale knowledge base for machine translation, Proceedings of the twelfth national conference on Artificial intelligence, 773-778 pp.

LTRL, (2011). <http://www.ucsc.cmb.ac.lk/ltrl/>, Language Technology Research Laboratory. UCSC, University of Colombo, Sri Lanka. Accessed 28 May 2014.

Madhura (2010), <http://maduraonline.com/> Madhura Online dictionary, Accessed 28 May 2014.

Swi-Prolog (2014), < http://www.swi-prolog.org/> Swi-Prolog Homw page , Accessed 20 May 2014.

WordNet (2013), < http://www.wordnet.lk/index.html> Sinhala Word Net, , Accessed 28 May 2014.

Wikipedai(2014).<http://en.wikipedia.org/wiki/Sinhala_ language>, Wikipedia, the free encyclopedia, Sinhala language, Accessed 10 June 2014.

BIOGRAPHY OF AUTHORS

[1]B Hettige is a PhD student of the Faculty of Information technology, University of Moratuwa, Sri Lanka. His research interests include Multi-agent technology, Machine translation and Sinhala Computing. He has produced more than 20 referred international and local publications to his credit.

[2]AS Karunananda is a Professor of Information Technology University of Moratuwa, Sri Lanka. At present he is the Dean of Research and Development of General Sir John Kotelawala Defence University. His research interests include Multi Agent Systems, Ontological Modelling, Machine Translation, and Theory of Computing.

[3]G Rzevski is a Visiting Professor in Multi-Agent Technology at Moratuwa University, Sri Lanka and Professor of Complexity Science and Design at the Open University, UK. His research interests are in Applications of Complexity Science and Multi-Agent Technology to a variety of practical problems including morphological, syntactical and semantic processing.