Multi-agent System Technology for Morphological Analysis

B. Hettige[1], A. S. Karunananda[2], G. Rzevski[3]
[123]A Faculty of Information Technology, University of Moratuwa, Sri Lanka,
[1]budditha@yahoo.com, [2]asoka@itfac.mrt.ac.lk, [3]rzevski@gmail.com.

Abstract-**Machine Translation involves multiple phases including morphological, syntax and semantic analysis of source and target languages. Despite there are numerous approaches to machine translations, handling of semantics has been an unsolved research challenge. We have been researching to exploit power of multi-agent Systems technology for machine translation by extending our rule-based machine translation system, BEES. Since there are no agent development framework specific to machine translation, our project has started by developing our own framework, MaSMT. This paper presents our research on the development of morphological analysis phase in MaSMT. Twenty-two ordinary agents and one manager agent have been implemented to model morphological analysis of English language. In contrast, MaSMT implements 206 agents and a manager agent to handle morphological analysis in Sinhala language. MaSMT has been developed in Java, while BEES is a Prolog implementation. Performance of morphological analysis by MaSMT and BEES has been evaluated. It was revealed that MaSMT performs much faster than BEES for morphological analysis of English sentences with a reasonable length such as 15 words. In case of Sinhala language too, MaSMT performs better than BEES. The difference in performances of MaSMT in Sinhala and English reflects the number of morphological rules in two languages. Due to parallel execution, MaSMT shows a significant improvement in identification of syntactic categories of words that have more than one interpretation. This feature will be reflected even better in syntactic and semantic analysis as they necessarily involve rules with multiple interpretations.**

1. Introduction

Machine Translation system is acomputer software that translates text or voice from one natural language into another with or without human assistance. In general, a machine translation system goes through several major steps including analysis of source language text, translation of syntactic categories of source into target language, and generation of the texts in the target language. Both the source language and target language sub systems are required to handle morphological, syntax and semantic aspects of the two languages. Knowledge required for language processing in those aspects can be represented as rules in the respective languages. As such, most of machine translation approaches are primarily based on rules. It is a known fact that for any given language it requires a considerable amount of rules for manipulation at its morphological, syntactic and semantic levels. For instance, we can identify almost 25 different rules to handle morphological analysis of words in English language, while Sinhala language uses 103 rules for handling morphological aspects of Sinhala words [1]. In this sense, morphological processing of a word requires to fire all such rules one after the other. Such a process is not only time consuming, but also leading to poor semantic interpretation upon the receipt of the first solution, which may not be the best. These issues have been encountered in many machine translation systems including the BEES project [2].

This research aims to improve the BEES by the existing features of multi-agent System (MAS) technology. More importantly, multi-agent system exploits the use of power of message passing among the agents that run in parallel to discover high quality solutions beyond the individual capacity of agents. We postulate that MAS technology can be used to model morphological, syntactic and semantic phases of a machine translation system.

In this paper, we present our research on the use of Multi-agent System technology for handling morphological aspects in English to Sinhala machine translation system. Despite there are hundreds of general purpose toolkits for the development of MAS, none of these have specialized for the domain of machine translation. Therefore, we have used our own MAS development environment, MaSMT, to develop the said machine translation solution. The English Morphological analyzer has been implemented with 22 ordinary agents and a manager agent to represent English morphological rules. In contrast, Sinhala Morphological analyzer has been developed with 206 ordinary agents a manager agent to represent Morphological rules in Sinhala language.

The rest of the paper is organized as follows. The section 2 describes some existing approaches for Morphological analysis. Section 3 reports multi-agent system technology for machine translation. Then section 4 briefly explains our novel approach for machine translation. Section 5 gives design of the Machine Translation system including brief description of each module of the English Morphological analyzer. After that, section 6 discusses how multi-agent based English Morphological analyzer works for the given set of words. The section 7 gives evaluation result and

section 8 reports conclusion and further works of the project.

2. Approaches to Morphological Analysis

Morphology is the identification, analysis and description of the structure of a given language's morphemes and other linguistic units, such as words, affixes, parts of speech etc. [5]
The morphological analyzer reads a word as an input and identifies the stems and affixes [6]. Then it returns complete morphological information about the given word. The term "Morphological analysis" in a language has a long history. The ancient Indian linguist Panini, formulated 3,959 rules of Sanskrit morphology. Historically this is regarded as the first attempt made for the morphological analysis recorded in the world. By using the Panini Sanskrit grammar Akshar and others have developed a Panini grammar model [7,8] for all Indian language families including, Hindi, Pali, Sanskrit, etc [8]. Although the number of researchers have already used this Panini grammar model to develop morphological analyzers for their language analysis.
Anusaaraka system has developed morphological analyzers for six Indian languages [9]. Anusaaraka has been designed to translate among major Indian languages and its morphological analysis is based on the paradigms of the Indian languages. The Paradigm is used both for word analysis as well as word generation. Also AksharBharati and others have already developed a Generic Morphological Analysis Shell that can be used to develop morphological analyzers for different minority languages [10]. This Shell uses finite state transducers (FST) with features to give the analysis of a given word. The generic Morphological Analysis Shell uses dictionaries, paradigm table and paradigm classes for its Morphological analysis.
Goyal and Lehal have already developed Morphological analyzer and generator for Hindi [11]. This Morphological analyzer has been developed through the paradigm approach and implemented with Windows based GUI. This project has been developed as part of the development of a machine translation system from Hindi to Punjabi Language.
Morphological analyzers for English language have been developed by many researchers. Among others, Koskenniemi's two-level morphology [12] was the first practical and most general model in the history of computational linguistics for the analysis of morphologically complex languages. Koskenniemi's Pascal implementation of morphological analysis was quickly followed by others. The most influential of them was the KIMMO system by LauriKarttunen [13] and his students at the University of Texas. PC-KIMMO is yet another morphological analysis tool, which was based on Koskenniemi's work and implemented in C. Among others, PC-KIMMO is supposed to be the only available free English morphological analyser with a wide coverage [14]. The lexicon used in PC-KIMMO considers verb, pronoun, noun, prepositions, adverbs and adjectives. The current version PC-KIMMO is implemented in C and can be run on a PC [15]. The PC-KIMMO accepts an input word from a user, and provides all possible morphological details of the word.

3. Multi-agent Systems Technology for Machine Translation

The multi-agent system technology is a modern approach for machine translation which is used to handle complex knowledge. In general multi-agent system contains four key components namely Multi-Agent Engine, Virtual world, Ontology and Interfaces [18]. The Multi-agent engine provides a run time support for agents. Virtual world is the environment of the multi-agent systems. The Ontology contains conceptual problem domain knowledge of each agent.

A. Existing MAS Development for Natural Language Processing

Considering the existing Natural Language Processing (NLP) approaches only few Multi-agent systems are available. Minakow and others [3] have developed a Multi-agentbased text understanding system for car insurance domain. This system uses Multi-agent system based approach to understand a given text. The system uses four steps to text understanding namely morphological analysis, syntax analysis, semantic analysis and pragmatic analysis. To analyze, the whole text is divided into sentences. Then first three stages are applied to each sentence. After analysing each paragraph, text is passed to pragmatic analysis.
Stefanini and others [4] have also developed a Multi-agent based general Natural language processing system named Talisman. The Talisman agents can communicate with each other without the central control. These agents are capable to directly exchange information using an interaction language. Linguistic agents are governed by a set of local rules. The TALISMAN deals with ambiguities and provides a distributed algorithm for conflict resolutions arising from uncertain information

B. Existing MAS Development frameworks

Frameworks save developer time and also aid in the standardization of Multi-agent System development. There are number of standard frameworks available for multi-agent system development including JADE, AgentBuilder, SeSAm etc.

JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language [19]. JADE is a middle-ware that complies with the FIPA specifications. This framework provides supporting GUI tools for debugging and deployment phases in the multi-agent development. In addition to the above, agent platform can be distributed across machines and the configuration can

be controlled via a remote GUI [20]. JADE successfully work with JRE 1.4 or above.

AgentBuilder [21] is an integrated software development tool that allows software developers with no intelligent agent technologies to quickly and easily build intelligent agent-based applications. AgentBuilder consists with three versions such as Lite, Pro and Pacs. AgentBuilder Lite provides several tools for development including Project Manager and Ontology Manager. The Ontology Manager provides tools for creating ontologies and automatic code generation using graphical object modelling tools. AgentBuilder Lite distributions are available for the Windows and Linux Platform.

SeSAm [22] (Shell for Simulated Agent Systems) provides a generic environment for modelling and experimenting with agent-based simulation. SeSAm provides tool for construction of complex models easily. SeSAm consist with several features and tools including visual agent modeling, Flexible environment and simulation analysis.

Agent based solution of the BEES project has been already developed to translate English text into Sinhala through the 9 agents [23][24]. These agents use existing rule-based translation module available in the BEES including English Morphological analyzer, English parser, Translator, Sinhala Morphological generator and Sinhala Sentence composer. As a result of that, core of the analysis is also rule-based in this previous system

4. A Novel Approach to Machine Translation

The approach behind MaSMT is based on the hypothesis that words employ as the building block of natural language understanding. This is valid for people who read a sentence word by word or otherwise by locating selected words such as nouns and verbs. Consequently, meaning of a sentence is determined by the interaction among words, which draw from all aspects of morphology, syntax and semantic, as appropriate. Thus MaSMT define words in a sentence as the agents. The agents pass messages among them within and across different level of analysis. As such, our machine translation approach is different from existing ones that sequentially define linguistic aspects such as morphological, syntax and semantics analysis.

For example, assume that we change our original understanding of a sentence when we read, say, from 5th word to the 8th word in the sentence. In such situations, we might continue to operate at the syntax level analysis or proceed to seek semantic information before completing the sentence at the syntax level. Thus syntax and semantic processing are inter-wound or parallel, not necessarily sequential. We argue that this connotation in language processing is much closer to how human beings process natural languages, and this can be effectively implemented by Agent technology

5. Design

MaSMT (Multi-agent System for Machine Translation) is a Java-based multi-agent development framework that can be used to develop Machine Translation applications. MaSMT provides two types of agents as ordinary agents and manager agents. A manager agent consists with number of ordinary agents within its control. Further, manager agents can directly communicate with other manager agents and the ordinary agents in the swarm that is assigned to a particular manager agent. Agents in a swarm can directly communicate only with the agents in the own swarm and the relevant manager agent. The framework primarily implements object-object communication, managing the agents (e.g. creation and execution), XML-based data passing and MySQL database connectivity for manipulation and use of the domain ontology. Agent communication in the framework has been implemented to comply with FIFA-ACL specifications [25].

This framework also provides MySQL database connector, XML database connector, message viewer and the agent monitor. Two connectors are used to communicate with MySQL database or XML database. A message viewer tool is used to view each message in the given message queue. This tool has been used to show message passing process in the MaSMT framework. The agent monitor is used to show each agent state (active, working, dead or busy).

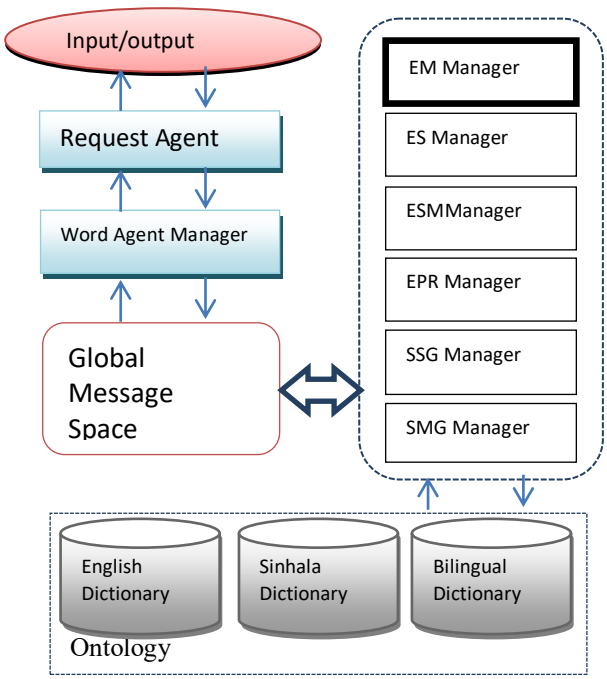

Figure1: Top level design of the Multi-agent based English to Sinhala machine translation system.

Through the MaSMT framework English to Sinhala Machine Translation system has been designed with 10 modules. Namely 7 managers, request agent, message space and the ontology. Word agent manager, English Morphological manager (EM),

English Syntax manager (ES), English Semantic manager (ESM), English pragmatic manager (EPR), Sinhala Syntax manager (SS) and Sinhala Morphological manager(SM) are the 7 managers of the system. Three lexical dictionaries namely, English dictionary, Sinhala dictionary, English-Sinhala Bilingual dictionary is used as ontology. Figure 1 shows top level design of the English to Sinhala machine translation system.

System reads English text as an input and provides translated Sinhala text as an output text. The request agent makes a request to translate the given text. Then word agent manager assign agents for each words in the given text. Therefore a word in the text is worked as an agent [23]. These word agents can communicate with each other agent(s). Further, message communication architecture has been designed through the MaSMT framework. The English Morphological manager controls all the functionalities on the English Morphological analysis. This Morphological processing system consists with 5 major components namely global message space, Morphological manager, Morphological agents, local message space and ontology. Figure 2 shows the design diagram of the Morphological processing system.

A. Morphological Manager

Morphological Manager manages its client agents. According to the MaSMT framework, each manager can fully control its client agents. Therefore, manager can create, remove or control its client agent(s) in the group. Manager agent creates all its clients automatically at the initialization stage. Morphological manager reads morphological rules which are available in the rule-base (part of the ontology) and assign each rule for a client agent. For instance, 85 grammar rules are available for Sinhala noun. Therefore 85 ordinary agents are created to implement the grammar). In addition to the above, manager can directly access each agent and send messages directly for its clients. The Manager agent reads input massages from the global message queue and provides relevant tasks for the client agents. Also Manager can control the priority of the agent and the stage of the clients. This facility removes the unnecessary work load from its client agents.

B. Morphological Agents

Morphological agents work under the control of the manager agent and each Morphological agent must have a manager agent. Morphological agent is a simple java based program (Thread) which support to do limited task(s). These Morphological agents can communicate with each other through the messages space and use peer-to-peer communication method. Morphological agent contains local message queue, Morphological rules and the ontology. Agents are responded for the messages which are available in its message queue. Each agent has been assigned for only the simple task and it response only for the assign task. Morphological agents response only two messages which contain "how are you" and "who am I". The agent reserved message "who are you" from the message queue then agent provides information about itself to the message sender by using message space. After reserving the message "who am I", it tries to do the Morphological analysis with support of the rules and its ontology.

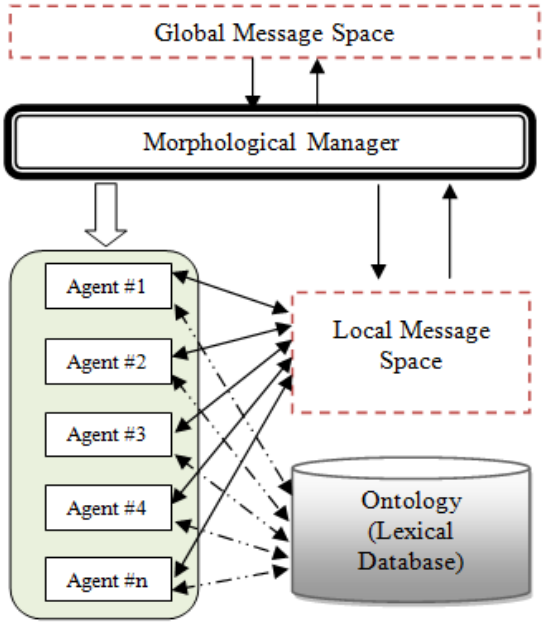

Figure2: design of the Morphological processing system

C. Local Message Space

Local message space is the message space for the each local agent group (Swarm). Each agent can directly communicate with each other through the local message space. This allows peer-to-peer connection. Morphological agent can send or receive messages from its local message space.

D. Global Message space

Global message space is the message space which can be used to access the system. This message space is visible only for the managers who are in the MaSMT framework. This message space uses for communication between managers. Agents can send messages only for the local messages and its managers.

E. Ontology

Ontology is the knowledge of each agent. English dictionary and the Sinhala dictionary are work as the ontology of the Morphological processing system. The English dictionary has been stored as a MySQL database. Further, 8 tables are used to store relevant lexical information for the English dictionary such as eng_reg_noun, eng_reg_verb, eng_reg_adjt, eng_irr_noun, eng_irr_verb, eng_irr_adjt, eng_irr_words and eng_pro_noun. To store morphological information system uses 2 tables namely eng_noun_mop_ruleandeng_verb_mop_rule.

Similarly there are another 8 tables to store relevant lexical information for the Sinhala dictionary such as sin_reg_noun, sin_reg_verb, sin_reg_adjt, sin_irr_noun, sin_irr_verb, sin_reg_advb,

sin_reg_prep and sin_reg_conj. To store morphological information system uses 3 tables namely sin_noun_mop_rule, sin_verb_mop_rule and sin_noun_case_rule These two dictionaries have been developed based on the existing prolog dictionaries available in the BEES project[26].

F. Messages

Messages have been used to communicate with each other. These messages are developed using FIPA ACL message stranded. 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.

Using the above design structure English and Sinhala morphological analyzers have been developed. The English Morphological analyzer has been implemented 22 ordinary agents and a manager agent for modeling morphological rules in English language. The English Morphological analyzer uses English dictionary as its Ontology. The English dictionary consists of more than 35000 English words including more than 20000 regular and irregular nouns, more than 10000 verbs and more than 5000 adjectives. In contrast, Sinhala morphological analyzer has been implemented 220 agents and a manager agent with regard to handling of nouns and verbs morphology in Sinhala language. The Sinhala Morphological analyzer uses Sinhala dictionary as its Ontology. The Sinhala dictionary consists with more than 80000 Sinhala words including 45000 nouns, 15000 verbs and 15000 other words.

6. How English Morphological Analyzer Works

This section describes how multi-agent based morphological analyzer works for a given English text. Figure 3 shows user interface of the English Morphological analyzer and figure 4 shows user interface of the Sinhala Morphological analyzer.

As a first step, the request agent makes a request to analysis. Then word agent manager reads the input text and word agents are automatically created for each word in the text. After that, each word agent ask message from English morphological agent "who am I". Word agent manager receives these messages from the word agents and send it to the English morphological manager through the global message space. The English morphological manager receives these messages and sends to its clients. (At this point each morphological agent has messages on their local message queue). The English morphological agents read these messages with the title "who am I" and try to analyze it with existing morphological rule. If rule is accepted then agents send relevant grammar information for the message sender through the English morphological manager. The English morphological manager sends these reply messages for the English word agent manager to deliver to its clients.

The Sinhala Morphological analyzer works with the same architecture of the English Morphological analyzer. The Sinhala Morphological agent also response the messages "how am I" and do the Sinhala morphological analysis.

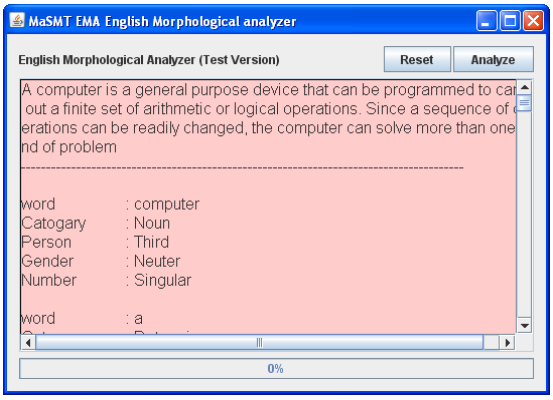

Figure3: User interface of the English Morphological analyzer

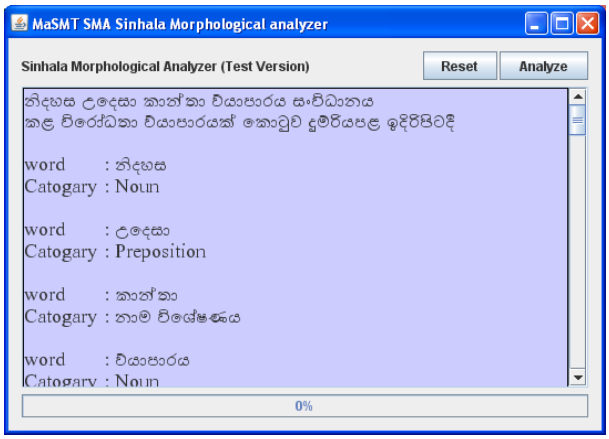

Figure4: User interface of the English Morphological analyzer

7. Evaluation

Two multi-agent based Morphological analyzers have been evaluated separatelythrough the existing evaluation methods which was used in the rule-based morphological analysis under the BEES project [27]. The English Morphological analyzer has been tested through the created test plan including 50 test cases and 100 sample words. The Sinhala morphological generator has been evaluated through the same evaluation method which is used to evaluate English Morphological analyzer. The Sinhala Morphological generator has been tested through the created test plan including 150 test cases and 300 sample words. Table 1 shows the evaluation results of the English and Sinhala morphological analysis

Further, two analyzers also tested against the rule-based morphological analyzer in BEES [16]. Figure 5

shows the experimental result of the rule based and Multi-agent based Morphological analysis. In the experiment we have separately calculated the time taken to analyze words in rule-base and multi-agent systems. The experimental result shows that, MaSMT performs much faster than BEES for morphological analysis of English sentences with a reasonable length such as 15 words. In case of Sinhala language too, MaSMT performs better than BEES. The difference in performances of MaSMT in Sinhala and English reflects the number of morphological rules in two languages.

Table 1: Evaluation Results of the English Morphological Analysis

Criteria	English Morphological Analyzer	Sinhala Morphological Analyzer
Success	92	260
Over-specified	-	-
Under-specified	5	28
Wrong Analysis	-	-
Over &under-specified	-	-
Irrelevant		
Not found	3	12
Total Solutions	100	300

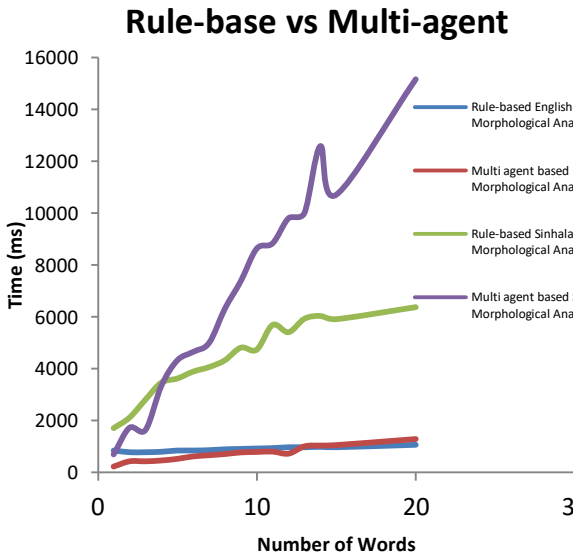

Figure5: performance comparison between rule-based vs. multi-agent approaches

8. Conclusion and Further works

This paper has reported our research on the use of MAS technology for handling morphological aspects in English to Sinhala machine translation. English and Sinhala morphological analyzers have been developed through our own MaSMT framework. MaSMT implements 22 ordinary agents and a manager agent for modeling morphological rules in English language. In contrast, MaSMT implements 206 agents and a manager agent with regard to handling of nouns and verbs morphology in Sinhala language. Two morphological analyzers have been tested through the created test plan. The experimental result shows 97% accuracy of the English morphological analysis and Sinhala morphological analyzer shows 96% accuracy. In addition to the above, both morphological analyzers have been compared with rule-based implementation in BEES. Experimental result shows that MaSMT performs much faster than BEES for morphological analysis of English sentences with a reasonable length

Due to parallel execution, MaSMT shows a significant improvement in identification of syntactic categories of words that have more than one interpretation. This feature will be reflected even better in syntactic and semantic analysis as they necessarily involves rules with multiple interpretations

Use of Multi-agent Systems technology to implement the other phases, namely, syntax and semantics analysis in machine translations has been considered as the key further work of MaSMT.

References

[1] B. Hettige, A. S. Karunananda, "Computational Model of Grammar for English to Sinhala Machine Translation", Proceedings of the International Conference on Advances in ICT for Emerging Regions - ICTer2011, Colombo, 2011.

[2] B. Hettige, (2011) BEES: Bilingual Expert for English to Sinhala, [online]. Available: http://dscs.sjp.ac.lk/~budditha/bees.htm

[3] I. Minakov and et al., "Creating Contract Templates for Car Insurance Using Multi-Agent Based Text Understanding and Clustering", Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems., HoloMAS:, 2007, pp 361-371.

[4] M. H. Stefanini, Y. Demazeau, "TALISMAN: A multi-agent system for natural language processing", In Proceedings of SBIA'95. - Springer Verlag:, 1995, pp. 312-322.

[5] Wikipedia,[online], Available: http://en.wikipedia.org.

[6] D. Jurafsky, J. H. Martin, "Speech and Language Processing", Boulder: University of Colorado, 2005.

[7] B. Akshar, V. Chaitanya, R. Sangal, "Natural Language Processing: A Paninian Perspective", New Delhi, India, Prentice Hall of India, 1995

[8] S. Bhate, S. Kak, "Panini's Grammar and Computer Science", Annals of the Bhandarkar Oriental Research Institute, vol. 72. 1993, pp. 79-94.

[9] Anusaaraka, [online]. Available: http://anusaaraka.iiit.ac.in/.

[10] B. Akshar, R. Sangal, D. M. Sharma,R. Mamidi, "Generic Morphological Analysis Shell", In Proceedings of LREC, 2004.

[11] V. Goyal, G. S Lehal, Hindi Morphological Analyzer and Generator, First International Conference on Emerging Trends in Engineering and Technology, IEEE, 2008 pp 1556-1559.

[12] K. Koskenniemi, "Two-level morphology: A general computational model for word-form recognition and production Publication", University of Helsinki, Department of General Linguistics, Helsinki, 1983.

[13] L. Karttunen, R. B. Kenneth, "Twenty-five years of finite-state morphology", In Inquiries into Words, Constraints and Contexts. CSLI Publications, 2005, pp71-83.

[14] K. Daneilf, Y. Schabes, M. Zaidel, D. Egedi, "A Freely Available Wide Coverage Morphological Analyzer for English", Proceedings of COLIN-92. 1992, pp. 23-28.

[15] L. Karttunen, R. B. Kenneth, "Twenty-five years of finite-state morphology", In Inquiries into Words, Constraints and Contexts. CSLI Publications, 2005, pp71-83.

[16] B. Hettige, A. S. Karunananda., "A Morphological analyzer to enable English to Sinhala Machine Translation", Proceedings of the 2nd International Conference on Information and Automation (ICIA2006), Colombo, Sri Lanka, pp 21-26, 2006.

[17] SWI-Prolog Home Page, [online] Available:http://www.swi-prolog.org/index.html.

[18] G. Rzevski, "A new direction of research into Artificial Intelligence", Sri Lanka Association for Artificial Intelligence 5th Annual Sessions.- 2008.

[19] JADE, [online], Available: http://jade.tilab.com/

[20] F. L. Bellifemine, G. Caire, D. Greenwood, Developing Multi-Agent Systems with JADE, John Wiley & Sons, Ltd, 2007.

[21] Agent Builder, URL: http://www.agentbuilder.com/Documentation/Lite/

[22] SeSAm, [Online], Available: http://www.simsesam.de/

[23] B. Hettige, A. S. Karunananda, "A Word as an Agent for Multi-agent based Machine Translation", Proceedings of the ITRU Research Symposium, Moratuwa, 2011

[24] B. Hettige, A. S. Karunananda, "Multi-Agent architecture for English to Sinhala Machine Translation", Proceedings of the 27th National IT conference (NITC10), Sri Lanka. 2010.

[25] FIPA:Agent Communication specifications, [online], Available: http://www.fipa.org

[26] B. Hettige, A. S. Karunananda., "Developing Lexicon Databases for English to Sinhala Machine Translation", proceedings of second International Conference on Industrial and Information Systems (ICIIS2007), IEEE, Sri Lanka, 2007.

[27] B. Hettige, A. S. Karunnanda., "An Evaluation methodology for English to Sinhala machine translation", Proceedings of the 6th International conference on Information and Automation foe Sustainability (ICIAfS 2010), IEEE., 2010.