ID 534

Three Address Code Based Semantics Processor for Sinhala

IWMHD Bandara[1], B Hettige[#1] and DDM Ranasinghe[2]

[1]*Department of Computer Engineering, General Sir John Kotelawala Defence University, Sri Lanka*
[2]*Department of Electrical and Computer Engineering, The Open University, Sri Lanka*

[#]budditha@kdu.ac.lk

Abstract—Semantic processing techniques have a wide interest in the field of Natural Language Processing. Processing a semantic from a natural language for human-machine communication is still a research challenge in this field. The Three-Address-Code is a type of intermediate code used by the compilers to identify the meaning of the source code or statements easily, with full accuracy. Therefore, the research captures the semantics of the Sinhala language text through this Three-Address-Code concept. This paper presents a Three-Address-Code based semantic processing system that can be used for human-machine communication using the Sinhala language. The proposed system comprises three components; namely Sinhala Part of Speech tagger, Sinhala chunker, and three-address-code based semantics generator. The system takes the Sinhala sentence as an input and generates the semantics information. This semantic processing system has been used under the PINA system for semantic processing.

Keywords: three-address-code, semantics, tagging, chunking, ontology

I. INTRODUCTION

Search engines have been using for several decades as a most important part of our digital surfing life. We are searching over billions of web resources to retrieve information from various resources. In the beginning search engines were lexical, that is they looked for matches of the query words without understanding the meaning of the query and only gives the links that contained exact query(Rajput, 2020). But with the NLP techniques it gives more hope towards understanding the semantics("Natural Language Processing - Semantic Analysis - Tutorialspoint," n.d.). To be think of humans are particularly good at conversational context and knowledge which make them easier to deal with ambiguity of words. But in the case of search engines when it comes to searches the problem is semantic search. So as a remedy for this problem here comes latest's insights form NLP research and resources. This can search for contents with same semantical phrases.

Natural language processing is a way to give the ability to understand natural languages for machines. Semantic processing is a sub part that is discussing in this; a way to get the meaning of a text. For that purpose, different techniques are used. A sentence has a logical concept of conveying the idea which can called as predicates. These can be identified by main verb or from the other parts of the sentence. Therefore, depending on the names given for these arguments, the method can be change. Semantic role labelling is the name given for identification of the predicate and the arguments for that predicate. There is also another kind of method called word sense disambiguation which is used by NLP for resolving various kinds of ambiguity("How Natural Language Processing will change the Semantic Web," 2016). A word can gives different meanings depending on the contextthey are being used. This makes the natural language understanding by machines more complex. These different meanings are called word senses. That is the sense of the word depends on Neighbor words around that word. Therefore, selecting the correct word sense is done by this method called word sense disambiguation. This method can have an impact on Machine translation processes, question answering and text classifications. There is another path called Named entity recognition in Natural language processing which focuses on identification of named entities such as persons, organizations, locations which can be denoted using proper nouns. The same words can represent different entities in different contexts. This method is used in text classification, content recommendations, trend detection etc. so these

are some existing semantic analysis methods used in Natural Language Processing.

In except for these methods our research is to use a compiler technique called three address code to extract the meaning of a given sentence and perform semantic analysis. Three address code technique is a compiler mechanism used my compilers to generate intermediate code which is form by separating the given expression into separate several instructions. These instructions or the intermediate code is then can be easily translate into assembly language which can machine understand. This method is used for generous sum of calculations. So, the research is carried out to find whether this method of calculations is possible to use in extracting meaning from a Sinhala Language context. The most important here is that whether the given context have uncertainty or not, the compilers can extract the meaning from the sentence. So, in here without considering ambiguity our main aim is to find out whether this same technique used by compilers can also be used to extract the meaning of Sinhala Language context.

The rest of the paper is structured as follows: A study on Semantic Processing is given in section 2. Section 3 discusses the Methodology and section 4 discuss How the system works and finally, section 5 will provide some concluding remarks.

II. SEMANTIC PROCESSING

The use of semantic processing is to find the exact meaning from a given text("Semantic Analysis In Linguistics," 2020). First part of the semantic analysis is to study the meaning off individual words of a given text and this is called as lexical semantics/processing. In the second part each word will be combined to provide meaningful sentences. A word can has several meanings based on the context of its usage in the sentence. Then they are ambiguous too, so we need to remove the ambiguity. For that we must check what is the prior sentence to that word. So, this sense we have word sentence disambiguation("NLP - Word Sense Disambiguation - Tutorialspoint," n.d.). Lexical ambiguity, syntactic and semantic are the problem that any NLP system must undergo. For that Part-of-Speech taggers can be used to solve the problem. What will happen when parsing is done when there is no proper meaning of the sentence, so in such cases we use semantic grammars to solve the problem("Part Of Speech Tagging – POS TaggingPOS (Part of Speech) NLP | byteiota," 2021). This grammar perform both syntactic and semantic checking.
Therefore, the extremely basic of semantic analysis is to get the proper meaning of the words or the sentence.

1.1. Meaning Representations

Semantic analysis use several approaches for the representation of meaning. They are,

- First order predicate Logic
- ML-based techniques; Semantic Nets
- Frames
- Conceptual Dependencies
- Rule-Based architecture

1.1.1. First Order Predicate Logic

First order logic is a flexible and computationally tractable meaning representation language. It provides a sound computational basis for the inference, expressiveness and verifiability and helps resolving ambiguity("Natural Language Processing - Semantic Analysis - Tutorialspoint," n.d.).

Eg: There is a teacher in town who teaches all children in town who do not have money themselves.

∃x (Teacher(x) ∧ In Town(x) ∧ ∀y (Children(y) ∧ InTown(y) ∧ ⊺ HaveMoney(y,y) à teachandlearn(x,y)))

1.1.2. ML-Based Techniques

English language is easy to identify and separate according to the meaning expressed by the text. If we consider about tokenization, it is about breaking a document or sentence into words for the ease of understanding the meaning. This is easier in English language. NLP rules are sufficient for this.
But what if we need to work with any other language except for English and here we use machine learning for tokenization. Using ML base techniques, we can train a model to identify and understand them.
There are 2 types, Supervised Machine Learning and Unsupervised Machine Learning for NLP and Semantic Analysis.

In Supervised machine learning a batch of documents are tagged with examples of what the system should looks like, and these documents are used to train the model. The larger the dataset its better as it learn more about the document.
Unsupervised Machine learning train a model without pre tagging. One feature that will focus here is Clustering. Clustering means grouping similar type of documents together into sets and then these sets of clusters are sorted based on their importance. This is called as hierarchical clustering.
Another feature is Latent Semantic Indexing. This technique identifies words and phrases which frequently occur.
Matrix Factorization is another technique, and it uses latent factors to break large matrix into combination of small matrices.
Unsupervised learning is tricky but data intensive than its supervised counterpart("Machine Learning (ML) for Natural Language Processing (NLP)," 2020).

1.1.3. Frames

Frame is a data structure to represent same properties for knowledge representations as in semantic networks. Semantic networks include nodes that show objects and explain the relationship between those objects. Frames have same ideas of inheritance and default values. They contain instructions to be understood well for computing things stored in other frames. Frames are useful in simulating commonsense knowledge which is exceedingly difficult for computers to handle("Framing And Frames In NLP," 2016). Frame based expert systems useful in representing knowledge organized by cause and effect. Frames are designed to show either generic or specific type of knowledge.

1.1.4. Conceptual Dependencies

Conceptual dependency is a way to represent the meaning of natural language sentences in a way that first shows the drawing inferences from the sentences. Is a theory of NLP which deals with representation of semantics of a language. It has been argued that representation in independent from the language which the sentences were originally had.Conceptual dependency argues that representation of a sentence is awaken not out of primitives related to the words in sentences but because of conceptual primitives that can be combined to form the meanings of the words in any language basis. In a dependency relation, one partner is dependent, and the other is dominant(Hull, 1972). The main goal of conceptual dependency base representations is to make explicit of what is implicit.

1.1.5. Rule-based Architecture

Rule base systems are used to store and process knowledge to interpret in a useful way. In logic we represent knowledge in a form of declarative static way. Rules in logic implicit what is true and false based on given conditions. Rule base systems are based on rules which say what to do in given conditions(Niu and Issa, 2014).A special type of interpreter controls when rules are invoked. Somehow these systems are very remarkably similar rules in logic. As for examples.

If it rains today the road will be wet today

Rains(today) à wet_road(today)

Therefore, a system whose knowledge base is represented as a set of rules and facts is known as a Rule Based system. It consists of IF-THEN rules, and collection of facts and interpreter controlling rules given the facts.

1.2. Semantic Analyzer and Intermediate Code Generation

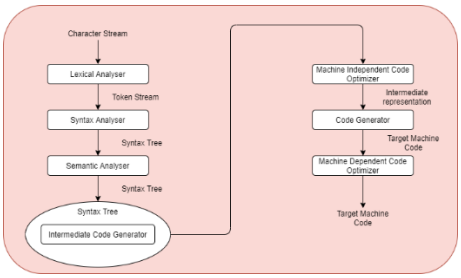


Figure 01: Intermediate Code Generation

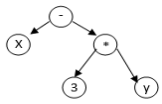
Intermediate code is The output from the parser and the input to the code generator.
There are 3 types of intermediate representations

- Abstract Syntax tree
- Postfix Notation
- Three Address Code

Abstract Syntax Tree

It depicts the natural hierarchical structure of a source program. A directed acyclic graph gives

the same information but in a more compact way as common sub expressions are identified.



X – 3 * y

Postfix Notation

Is a linearized representation of a syntax tree. There are list of nodes of the tree in which a node appears after its children immediately.

As these types of representations are present but they never used for Sinhala Language semantic processing. Therefore, Three Address Code technique is never used for processing semantics.

III. APPROACH

Three Address Code

Three address code technique is used in programming languages to process unambiguity in sentences. Compilers use three adress code to identify the absence of Unambiguity in the sentences. Therefore, the research is focused on use of Three Address Code on Semantic processing for Sinhala Language.

Three Address Code is an abstract form of intermediate code. The way to implement is as records with fields with the operator and the operands. At most it has three addresses in the instruction and one operator on the right hand side of the assignment.

General statement form: x = y op z

Longer expressions are simplified into small expressions.

Eg: z = x – 2 * y

t1 = 2 * y t2 = x - t1 z = t2	or	t1 = 2 * y z = x - t1

The contents arg1, arg2 and result are pointers to the symbol table entries for the names represented by those fields.

There are 3 representations of Three Address Code.

1. Quadruples

A quadruple has 4 fields: op, arg1, arg2, result.

For example, the three address instruction X= y+x à + is op, y is arg1, x is arg2, X is result.

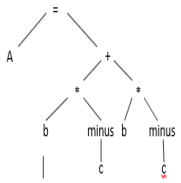
t1 = minus c
t2 = b*t1
t3 = minus c
t4 = b*t3
t5 = t2 +t4
a = t5

(Three Address Code)

op	Arg1	Arg2	result
minus	c		t1
*	b	t1	t2
minus	c		t3
*	b	t3	t4
+	t2	t4	t5
=	t5		a

2. Triples

Has only 3 fields; op, op1, op2



(Syntax Tree)

op	Arg1	Arg2
minus	c	
*	b	(0)
minus	c	
*	b	(2)
+	(1)	(3)
=	a	(4)

(Triples)

Figure 2

3. Indirect Triples

Indirect triples consist of a listing of pointers to triples.

Intermediates codes are closed to machine instructions but machine independent. The given program in its source language is transform into equivalent program by intermediate code generator. Intermediate language can be many different languages and the compiler decides the intermediate language. As described above Syntax trees, Postfix notation and Three address code can be used as an intermediate language(“Compiler - Intermediate Code Generation - Tutorialspoint,” n.d.).

Applying Three address code for Semantic processing in Sinhala is shown in the section below.

IV. DESIGN & IMPLEMENTATION

L. The architecture of the system

The following figure describes the entire design process of the suppose system. The system itself consists of 4 modules: Tokenizer, Sinhala POS Tagger, Sinhala phase base chunker and Ontology Generator.

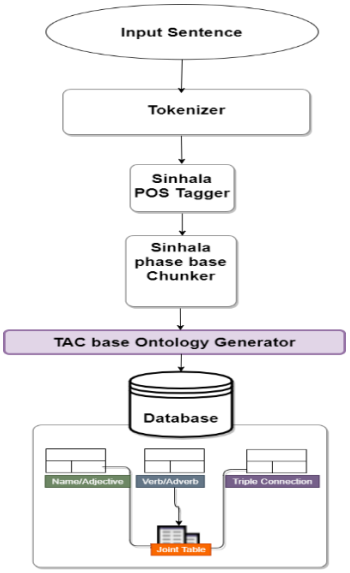


Figure 3: System Design

Module 1: Tokenizer

This module will focus on extracting words from a given sentence.

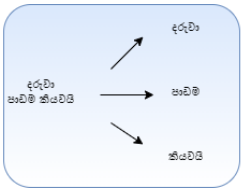


Figure 4: Tokenizing

The very first tasks performing in computing is working with textual input. Therefore, it is required to transform the textual representation suitable for computational processing. This process of transforming into a something suitable for computation is called as lexical analysis or tokenization(Pigulla, 2009). Tokenization is hard because in grammatical context it has lot of issues to address and lots of ambiguities to resolve (Zubac and Dadarlat, 2013). In (Dale et al., 2000) states that tokenization must address language dependency, character dependency, application dependency and corpus dependency. The most important section word related ambiguity concerns abbreviations, a acronym, multi part word expressions (Zubac and Dadarlat, 2013).

Module 2: Sinhala POS Tagger

Through this module it will assign POS tags for each word which are already separated from the input sentence. Tags are assigned to identify each form of the word whether its is a noun, verb, adjective, adverb and so on.

So, the system will identify the words by its tags.

දරුවා**_NNP** පාඩම**_NNP** කියවයි**_VFM**

In traditional Sinhala grammar variations have been proposed for Part of speech. This is because of the existence of several grammatical schools in Sinhala Language. (Weerasinghe et al., 2009). Therefore depending on those classifications a Sinhala POS tag set was already implemented (Fernando et al., 2016).

Module 3: Sinhala Phase base Chunker

This module will identify the sentence in its noun or verb phrase forms. Once the tagging part is done through the previous module the chunker will look through those tags and define what form of phrases that they are belong. From the patterns the rules are learned by considering the neighbor word and POS tags the phrase identification was done (Sangal et al., 2007).

Module 4: Ontology Generator

This module is the final and most innovative section which is focus through this project. Here once identifying the verb and noun phrases the words in preprocessed sentences will be stored to an ontology and this extraction is done through the compiler based mechanism called Three Address Code technique which is described above.

Figure5 : Applying Three Address Code

The above figure shows how the three address code is used to extract the meaning from semantics.

The ontology will contain number of tables depending on the situation (noun table , verb table, adjective table…). And most importantly Triple Connection table. This table will maintain the original connection (i.e. where those words are come from) of the words which are now stored in the ontology. This will help to traceback the system when generating answer for a question which is exactly the reversed way of doing the same process described till now.

Therefore, through all these 4 modules the system will learn and through the knowledge it gains, the system will be able to answer any question within its domain of knowledge in Sinhala Language.

V. HOW THE SYSTEM WORKS

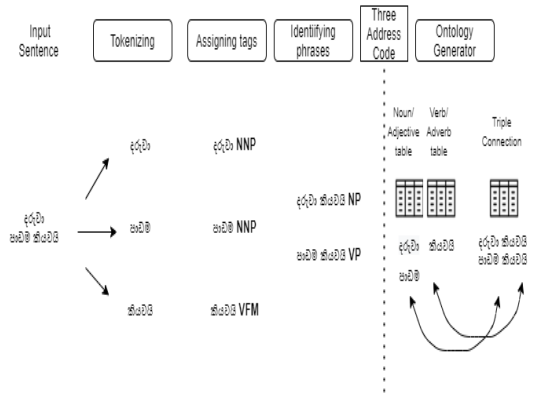


Figure 6: Working of the System

Figure 5 shows how each module of the system is contributing to the system. First the sentences in text form will be entered to the system as the input. Then the tokenizer will extract the words from the sentence separately. While gaining the original words from the input sentence it will also define every form of each word. Because in Sinhala Language the context is very much larger when comparing with English. As you already know the syntax of the Sinhala is very much differ from English.

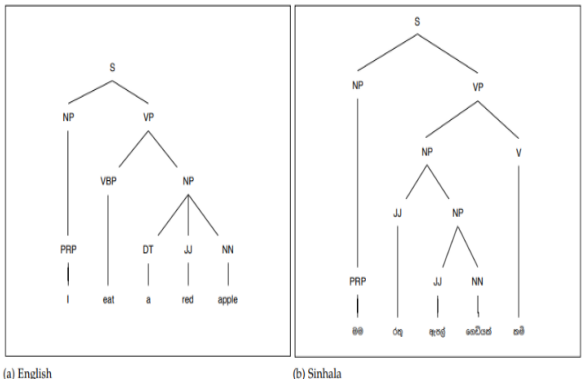


Figure 7: English syntax vs Sinhala syntax

Therefore, the order of the words when defining some context in Sinhala is not like that in English. After tokenizing the words the POS tagger will assign tags to each word after identifying its form of context.

Tag	Description
DET	Determiner
NNN	Common Noun Neuter
NNC	Common Noun
NNJ	Adjective Noun
NNR	Common Noun Root
NNS	Noun Plural
NNV	Sentence Ending
NNP	Proper Noun Singular
NNPS	Proper Noun Plural
NNM	Common Noun Masculine
NNF	Common Noun Feminine
PDT	Predeterminer
RB	Adverb
RRPCV	Particle in Compound Verbs
PRPC	Pronoun Common
PRPF	Pronoun Feminine
PRPN	Pronoun Neuter
SVCV	Supportive Verb in Compound Verb
VP	Verb Participle
PUNC	Punctuations
FS	Full Stop
NUM	Number
PRP	Pronoun Common
VNF	Verb Non Finite
POST	Postpositions
VNN	Verbal Noun
NVB	Noun in Kriya Mula
JVB	Adjective in Kriya Mula

Figure 8: POS tags

The above figure shows some of Part of Speech tags that are going to use in this system. After assigning tags the chunker will identify the grammatical context of the sentence. While referring to the tags it already assigned, the chunker will define each as noun phrase or verb phrase. Then after going through all these processes the Three Address Code technique will be applied to the produced output and the words so extracted will be stored into the database. In this way the system will record every input it gets through in the form of text. The same process will be reversed back to generate the answer when some question is asked through the learned knowledge by the system itself. In this case the previously explained Triple connection table is more beneficial since it contains the real connection of each word that are separately stored into the ontology.

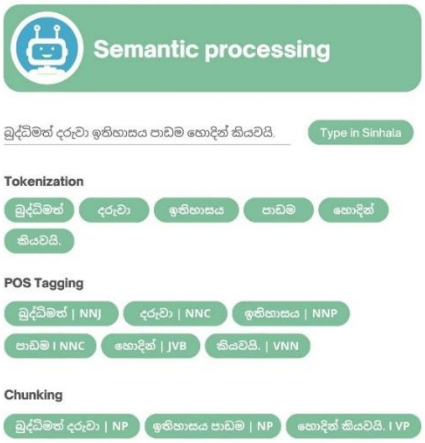


Figure 9: Sample GUI

VI. DISCUSSION AND CONCLUSION

Sinhala is the native language of Sri Lanka. Usually about over 16 million odd people are

using this language. The derived alphabet consists of vowels, consonants, semi consonants, and conjunction consonants. "In Sinhala alphabet there are 18 vowels, with 8 stops, 2 fricatives, 2 affricates, 2 nasals, 2 liquids and 2 glides as well as 41 consonants"(Rajamanthri, 2021). As well as other languages Sinhala language also have a set of grammar rules; Sinhala uses postposition instead of prepositions. Which is opposite to the way of writing of English. Common order for Sinhala context is subject, object and finally the verb. According to grammar the verb should behave with respect to noun. This context is depending on so many facts like gender of the verb, singular/plural, and the tense. This is applied for written Sinhala and in spoken Sinhala this is not very much applicable. That is whatever its gender or singular or plural it only depends upon the tense of the sentence. Therefore, by considering all above facts defining a semantic processor for Sinhala using NLP techniques is much more complex. Sinhala is a low resource language for which linguistic tools are not been properly defined. Therefore, when implementing NLP base tools, we must rely on language independent techniques(Ranathunga and Liyanage, 2021). And another problem arises here is its difficult to identify the dialect. As a solution to this problem some researchers have developed a library package which can convert spoken Sinhala words to Sinhala text called "Sphinx"(Gunarathne et al., n.d.).

This paper has reported the design of new semantic processing technique namely Three Address base Semantic Processor for Sinhala. The system is specific to develop process semantics in Sinhala using a compiler base mechanism which is not in use in existing semantic processing techniques. The system provides 4 modules to perform each task explained in entailed above and give an accurate output. The project lies on both the areas; expert systems and natural language processing and after doing thorough research on existing semantic techniques we used to have this mechanism to apply for semantic processing.

REFERENCES

Compiler - Intermediate Code Generation - Tutorialspoint [WWW Document], n.d. URL https://www.tutorialspoint.com/compiler_design/compiler_design_intermediate_code_generations.htm (accessed 6.20.21).

Dale, R., Moisl, H., Somers, H., 2000. Handbook of Natural Language Processing. CRC Press.

Fernando, S., Ranathunga, S., Jayasena, S., Dias, G., 2016. Comprehensive Part-Of-Speech Tag Set and SVM based POS Tagger for Sinhala, in: Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016). The COLING 2016 Organizing Committee, Osaka, Japan, pp. 173–182.

Framing And Frames In NLP, 2016. . Practical NLP. URL https://nlppod.com/framing-commonly-used-frames-nlp/ (accessed 6.20.21).

Gunarathne, W., Ramasinghe, T., Wimalarathne, D., Balasuriya, B., Hettige, B., n.d. Sinhala Speech to Text Library using Sphinx 6.

How Natural Language Processing will change the Semantic Web [WWW Document], 2016. . SEMANTiCS 2020 US. URL https://2020-us.semantics.cc/how-natural-language-processing-will-change-semantic-web (accessed 6.20.21).

Hull, B.W., 1972. Conceptual dependency structure in the NLP natural language processor.

Machine Learning (ML) for Natural Language Processing (NLP) [WWW Document], 2020. . Lexalytics. URL https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing (accessed 6.20.21).

Natural Language Processing - Semantic Analysis - Tutorialspoint [WWW Document], n.d. URL https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_semantic_analysis.htm (accessed 6.20.21a).

Niu, J., Issa, R.R.A., 2014. Rule-Based NLP Methodology for Semantic Interpretation of Impact Factors for Construction Claim Cases, in: Computing in Civil and Building Engineering (2014). Presented at the 2014 International Conference on Computing in Civil and Building Engineering, American Society of Civil Engineers, Orlando, Florida, United States, pp. 2263–2270. https://doi.org/10.1061/9780784413616.281

NLP - Word Sense Disambiguation - Tutorialspoint [WWW Document], n.d. URL https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_word_sense_disambiguation.htm (accessed 6.20.21).

Part Of Speech Tagging – POS Tagging in NLP | byteiota, 2021. URL https://byteiota.com/pos-tagging/ (accessed 6.20.21).

Pigulla, R., 2009. Enhancing Text Tokenization with Semantic Annotations using Natural Language Processing and Text Mining Techniques.

Rajamanthri, L., 2021. Sinhala Language Plagiarism Tool with Internet Resources Using Natural Language Processing.

Rajput, A., 2020. Semantic Search Engine using NLP [WWW Document]. Medium. URL https://medium.com/analytics-vidhya/semantic-search-engine-using-nlp-cec19e8cfa7e (accessed 6.20.21).

Ranathunga, S., Liyanage, I.U., 2021. Sentiment Analysis of Sinhala News Comments. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 20, 59:1-59:23. https://doi.org/10.1145/3445035

Sangal, R., Bendre, S., Sharma, D.M., Mannem, Prashanth Reddy, Bharati, A., Mannem, Prashanth R., 2007. Organizers.

Semantic Analysis In Linguistics: An Introduction [WWW Document], 2020. . MonkeyLearn Blog. URL https://monkeylearn.com/blog/semantic-analysis/ (accessed 6.20.21).

Weerasinghe, R., Herath, D., Welgama, V., 2009. Corpus-based Sinhala Lexicon, in: Proceedings of the 7th Workshop on Asian Language Resources (ALR7). Association for Computational Linguistics, Suntec, Singapore, pp. 17–23.