



Addis Ababa University  
College of Natural Sciences

Amharic Sentence Generation from Interlingua Representation

Kibrewossen Yitbarek

A Thesis Submitted to the Department of Computer Science in Partial  
Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

December 2016

Addis Ababa University  
College of Natural Sciences

Kibrewossen Yitbarek

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by Kibrewossen Yitbarek, titled: *Amharic Sentence Generation from Interlingua Representation* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	<u>Name</u>	<u>Signature</u>	<u>Date</u>
Advisor:	<u>Yaregal Assabie (PhD)</u>	_____	_____
Examiner:	<u>Mesfin Kifle (PhD)</u>	_____	_____
Examiner:	<u>Mulugeta Libsie (PhD)</u>	_____	_____

## **Abstract**

Sentence generation is a part of Natural Language Generation (NLG) which is the process of deliberately constructing a natural language text in order to meet specified communicative goals. The major requirement of sentence generation in a natural language is providing full, clear, meaningful and grammatically correct sentence. A sentence can be generated from different possible sources, including a representation which does not depend in any human languages, which is an Interlingua. Generating a sentence from an Interlingua representation has numerous advantages. Since Interlingua representation is unambiguous, universal and independent of both the source language and the target language, the generation should be target language-specific, and likewise should be the analysis. Among the different Interlinguas', Universal Networking Language (UNL) is commonly chosen in view of various advantages over the other ones. Various works have been done so far for different languages of the world to generate sentences from UNL expression but to the best of our knowledge there are no works done so far for Amharic language.

In this thesis, we present Amharic sentence generator that automatically generates Amharic sentence from a given input UNL expression. The generator accepts a UNL expression as an input and parses to build a node-net from the input UNL expression. The parsed UNL expressions are stored in a data structure which could be easily modified in the successive processes. UNL-to-Amharic word dictionary is also prepared and it contains the root form of Amharic words. The Amharic equivalent root word and attributes of nodes in a parsed UNL expression will be fetched from the dictionary to update the head word and attributes of the corresponding node. Then, the translated Amharic root words will be locally reordered and marked based on the Amharic grammar rules. When the nodes are ready for generation of morphology, the proposed system makes use of Amharic morphology data sets to handle the generation of noun, adjective, pronoun, and verb morphology. Finally, the function words are inserted to the morphed words so that the output matches with a natural language sentence.

The evaluation of the proposed system has been performed on dataset of 142 UNL expressions. Subjective tests like adequacy and fluency tests have been performed on the proposed system. Moreover, the quantitative test or error analysis has also been performed by calculating Word Error Rate (WER). From this analysis, it has been observed that the proposed system generates 71.4% sentences that are intelligible and 67.8% sentences that are faithful to the original UNL expression.

Consequently, the system achieved a fluency score of 3.0 (on a 4-point scale) and adequacy score of 2.9 (on a 4-point scale). Furthermore, the proposed system has word error rate of 28.94%. These scores of the proposed system can be improved further by improving the rule base and lexicon.

**Keywords:** Natural Language Generation, Interlingua, Universal Network Language, Universal Word, Head Word, Attribute, Local Reordering, Morphology Generation, Fluency, Adequacy.

## **Acknowledgments**

With the grace of God, it is very special moment to acknowledge the contributions of those who supported me during the journey to achieve this task. This research work comes out of very valuable guidance from my advisor Dr. Yaregal Assabie. He has always been there to guide me towards the right direction whenever I got stuck with ideas. I want to thank him for his valuable evenings and week-ends that he spent with me in finalizing the writings of this thesis work.

I would also like to thank the experts who were involved in preparing dictionary and grammar rules for this thesis: Mrs. Firehiwot Seifu, Mr. Negesu Tsegaye, Ms. Rosa Gizaw and Mr. Yoseph Ayele. Without their passionate participation and input, the validation survey could not have been successfully conducted.

I would also like to acknowledge Mr. Daniel Sahle and Mrs. Edil Endalew as the second reader of this thesis, and I am gratefully indebted to them for their very valuable comments on this thesis.

My sincere thanks also goes to Dr. Mulugeta Libsie and Dr. Mesfin Kifle for their constructive comments and suggestions for the improvement of this work.

Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Above all, the almighty God, for answering my prayers and for giving me the strength to be able to perform this study.

Thank you.

## Table of Contents

List of Figures .....	iv
List of Tables .....	v
List of Algorithms .....	vi
List of Listings .....	vii
Acronyms .....	viii
Chapter One: Introduction .....	1
1.1 Background .....	1
1.2 Motivation .....	3
1.3 Statement of the problem .....	4
1.4 Objectives .....	4
1.5 Methodology .....	5
1.6 Scope and Limitations .....	5
1.7 Application of Results .....	5
1.8 Organization of the Rest of the Thesis .....	6
Chapter Two: Literature Review .....	7
2.1 Natural Language Generation .....	7
2.2 Interlingua Representation .....	9
2.2.1 Interlingua Definition and Components .....	9
2.2.2 Interlinguas in Machine Translation .....	10
2.2.3 UNL Expression .....	14
2.3 Amharic Grammar .....	23
2.3.1 Phrase Construction .....	25
2.3.2 Sentence Construction .....	26
2.4 Morphological Generator .....	29

2.5	Summary .....	30
Chapter Three: Related Work .....		31
3.1	UNL Based Sentence Generation.....	31
3.2	Application of UNL .....	32
3.3	Summary .....	35
Chapter Four: Amharic Sentence Generator.....		37
4.1	Architecture of Amharic Sentence Generator .....	37
4.2	UNL parser.....	39
4.3	Term Translation .....	42
4.4	Local Reordering.....	44
4.4.1	Matrix Based Priority of Relations` .....	45
4.4.2	Amharic Marking Order .....	48
4.4.3	Reordering for simple sentences.....	52
4.4.4	Local Reordering for Compound Sentence.....	57
4.4.5	Special cases in Local Reordering .....	59
4.5	Amharic Morphology Generator .....	63
4.5.1	Noun morphology .....	65
4.5.2	Verb Morphology.....	68
4.5.3	Function Word Insertion .....	71
4.6	Summary .....	74
Chapter Five: Experiment .....		76
5.1	Prototype Development.....	76
5.2	Evaluation.....	77
5.2.1	Selection of Sentences for Test Cases .....	77
5.2.2	Evaluation of Amharic Sentence Generator .....	77

5.2.3	Tests for the evaluation.....	78
5.3	Discussion .....	79
5.3.1	Results for fluency score.....	80
5.3.2	Results for Adequacy Score.....	83
5.3.3	Results to the Error Analysis .....	84
Chapter Six: Conclusion and Future Work.....		86
6.1	Conclusion.....	86
6.2	Contribution of the Research.....	87
6.3	Future work .....	88
References.....		89
7	Annexes .....	94
Annex A: UNL Relations.....		94
Annex B: Priority Matrix of UNL Relations for Amharic .....		106

# List of Figures

Figure 2.1: Interlingua MT System Architecture.....	10
Figure 2.2: Interlingua-based Translation among n Languages.....	11
Figure 2.3: A UNL System .....	14
Figure 2.4: UNL Graph for Example Sentence Given In Listing 2.2 and/or Listing 2.3 .....	22
Figure 4.1: Architecture of Amharic Sentence Generator .....	38
Figure 4.2: A UNL Graph Representation for UNL Binary Relation rel(UW1, UW2) .....	40
Figure 4.3: Simplified UNL Graph for UNL Expression given in Listing 4.1 .....	42
Figure 4.4: UNL Graph of Two Nodes with Same Parent.....	46
Figure 4.5: Priority Matrix for Reordering of UNL Expression Given In Listing 4.1 .....	47
Figure 4.6: Priority matrix of Listing 4.5.....	49
Figure 4.7: UNL Graph of Listing 4.5 .....	49
Figure 4.8: Amharic Verb Inflection Structure.....	51
Figure 4.9: Inflection Marking Order .....	52
Figure 4.10: UNL Graph for UNL Expression in Listing 4.10.....	54
Figure 4.11: Priority Matrix for ‘agt’ and ‘obj’ Relations .....	55
Figure 4.12: UNL Graph with Scope Node for Listing 4.12 .....	58
Figure 4.13: Priority Matrix for ‘agt’, ‘tim’ and ‘plc’ Relations .....	59
Figure 4.14: UNL Graph for UNL Expression Listing 4.13.....	61
Figure 4.15: UNL Graph Having Relation ‘or’ for UNL Expression Listing 4.15 .....	63
Figure 5.1: Amharic Sentence Generation GUI.....	76
Figure 5.2: Distribution of adequacy scores for various fluency score. ....	84
Figure 5.3: Word Error Rate Analysis of the Proposed System .....	85

# List of Tables

Table 2.1: Description of UNL Attributes .....	17
Table 2.2: Syntax of UNL Sentence in Table Format .....	19
Table 2.3: Syntax of UNL Sentence in List Format .....	19
Table 2.4: Description of Word Dictionary Entries .....	22
Table 4.1: Matrix Representation for (N1 L N2) Structure .....	46
Table 4.2: Sample Noun Morphology Rule base.....	66
Table 4.3: Sample Parameter Value Passed to the Amharic Morphology Generator.....	69
Table 4.4: Sample Rule base for Verbs in Amharic. ....	70
Table 5.1: Four Point Scale of Fluency Score .....	78
Table 5.2: Four Point Scale of Adequacy Score.....	79
Table 5.3: Generated Amharic Sentence from UNL Expression Sample .....	80

# List of Algorithms

Algorithm 4.1: UNL Parser.....	41
Algorithm 4.2: Reordering Algorithm for Simple Sentences .....	53
Algorithm 4.3: Reordering Algorithm for UNL graphs with scope node.....	57
Algorithm 4.4: Algorithm for Special Case Relations.....	60
Algorithm 4.5: Processing of Noun Morphology Rule base.....	68

# List of Listings

Listing 2.1: UW ‘run’ Restricted with Constraint List .....	16
Listing 2.2: UNL Expression Table Format Example .....	20
Listing 2.3: UNL Expression List Format Example .....	21
Listing 2.4: Amharic Pronoun Example .....	24
Listing 4.1: UNL Expression Example Sentence .....	38
Listing 4.2 : Node List Created By UNL Parser for UNL Expression Given In Listing 4.1 .....	42
Listing 4.3: Amharic Dictionary Sample Entries.....	44
Listing 4.4: Updated Node list form Listing 4.2.....	44
Listing 4.5: UNL Expression Example to Illustrate Marking.....	48
Listing 4.6: Node List of Listing 4.6 after Term Translation is Processed.....	49
Listing 4.7: Node List of Listing 4.6 after Morphology Generation is Processed.....	50
Listing 4.8: Node List of Listing 4.6 after Marking of HWs is Processed .....	51
Listing 4.9: Node List of Listing 4.6 after Morphology Generation is Processed.....	51
Listing 4.10: UNL Expression Example to Illustrate Simple Sentence Reordering.....	54
Listing 4.11: Node List of Listing 4.10 after Term Translation is Processed.....	55
Listing 4.12: UNL Expression Example to Illustrate Compound Sentence Reordering .....	58
Listing 4.13: UNL Expression Having Relation ‘and’ .....	60
Listing 4.14: Node List of Listing 4.13 after Term Translation is Processed.....	61
Listing 4.15: UNL Expression Having Relation ‘or’ .....	62
Listing 4.16: UNL Expression Example 2 Sentence .....	64
Listing 4.17: Node List of Listing 4.4 after Morphology is Processed.....	73
Listing 4.18: Node List of Listing 4.17 after Function Word Insertion is Processed .....	74
Listing 4.19: Function Word Insertion Sample Rules7 .....	74

# Acronyms

DLT	Distributed Language Translation
DS	Dialogue Systems
HW	Head word
IE	Information Extraction
KANT	Knowledge-based, Accurate Natural-Language Translation
LCS	Lexical Conceptual Structure
LIS	Library Information System
MT	Machine Translation
NLG	Natural Language Generation
NLP	Natural Language Processing
QA	Question Answering
SIGGEN	Special Interest Group in Natural Language Generation
SOV	Subject Object Verb
TAM	Tense-Aspect-Mood
TS	Text Summarization
UNDL	Universal Networking Digital Language
UNITRAN	UNIversal TRANslator
UNL	Universal Network Language
UW	Universal Word
WER	Word Error Rate

# Chapter One: Introduction

## 1.1 Background

In our everyday life, we frequently exercise sentence generation in order to communicate with others. Of course, communication cannot be accomplished successfully just because we have the ability to transmit our thoughts. Understanding what others are trying to say is also a key point for a successful communication. Both concepts are equally important in Natural Language Processing (NLP), however with precisely unknown reasons, less focus is given to sentence generation [1].

Sentence generation is a computational process of producing sentences automatically in some human language with a specific communication intention. Its purpose is producing text which satisfies the communicative goal of the speaker or the writer. Sentence generation is a part of Natural Language Generation (NLG) which is a computerized process of analyzing and converting data into meaningful information [2]. The major requirement of sentence generation in a natural language is providing full, clear, meaningful and grammatically correct sentence. An NLG system converts a computer based representation into a natural language representation. Their main target is to make decisions about how to put a concept into words.

Generally, we can generate a sentence from different possible inputs. One can generate a sentence from an event occurred; we may find such systems in the areas of annotation. Others can produce a sentence from a sentence in other language; we can find such systems in the areas of machine translation. Also, a sentence can be generated from a representation which does not depend in any human languages which is an Interlingua representation.

For this research, we proposed to use Interlingua representation. The reasons for choosing Interlingua representation are the following. First, it is unambiguous and it represents the full content of the input text. In addition, it is universal, capable of representing the abstract meaning of any text, belonging to any domain or language. Also, it represents the content of the input alone and not be influenced by the formal representation of the content in the source language text. Furthermore, it is independent of both the source language and the target language; analysis should be source language-specific and not oriented to any particular target language, and likewise should be the generation [3].

Unless the generated text has similar semantic meaning with the information which the writer or speaker wants to express, there will still be a huge communication problem. NLG has an opposite meaning with Natural Language Understanding. NLG systems always have to grab the semantic meaning of the sentence to be generated. Therefore, understanding the semantic meaning of the information which is going to be communicated must be understood by the system. That means we cannot simply generate text without having a mechanism to make the system understand what the speaker or writer wants to say. NLG can be applied in different areas of natural language processing like Dialogue System, Machine Translation, Speech-to-Text translation, and so on.

When the sentence is created, it always focuses on the goal of communication. Here, one of the major things we need to be concerned during the generation of a natural language text is the text must not be redundant and should be meaningful to those who read the output. Even if natural languages are different in their nature, the generation process shares similar stages. The stages as proposed by Dale and Reiter [2] are:

- **Content determination:** focus on deciding what information to mention in the text.
- **Document structuring:** focus on overall organization of the information to convey.
- **Aggregation:** focus on merging of similar sentences to improve readability and naturalness.
- **Lexical choice:** focus on putting words to the concepts.
- **Referring expression generation:** focus on creating referring expressions that identify objects and regions.
- **Realization:** focus on creating the actual text which should be correct according to the rules of syntax, morphology, and orthography.

Having this in mind, we also have to make sure that the sentence generated has the appropriate grammatical structure and meaningful message for the native speakers of the language. In order to generate a sentence, which is precise and grammatically correct, we need to have a technique that enables us to understand the real and precise meaning of each word that could possibly be used in a sentence.

Although the generation problems are as sophisticated as those of the analysis, Amharic generation received no attention compared to Amharic morphological and syntactic analysis. In the event of translating a sentence from one language to the other, identifying what the speaker really wanted

to say is a major problem. In addition, natural language generated from a natural language by machine has a problem of redundancy and impreciseness [4]. However, an Interlingua representation provides content description which does not depend on any natural language. Putting Interlingua representation into action will solve the major issue in NLG; since it is based on speaker or writer intention rather than literal meaning [5].

**For example:**

1) X: greeting (greeting=good morning) →Interlingua Representation.

እንደምን አደርክ(ሽ/ሩ/አቸሁ):: →The possible sentence generation in X representation

2) Y: give\_information + plan + trip (who=I, visit-spec=(identifiability=no, vacation, time =(season=( identifiability =non-distant, summer)), location= name- Harar))

በዚህ ከረምት ሐረር ሂዶ የመዝናናት እቅድ አለኝ::

Other languages like Arabic, French, Hindi and the like have implemented text generation from Interlingua representation. If we consider the Arabic language most of their research are conducted on sentence generation using Interlingua based method [6, 7]. India also made a progress in generating text from Interlingua [4].

This thesis aims to address the current problems towards generation of Amharic sentence from Interlingua representation and provides a framework for Amharic sentence generator which can generate a meaningful and precise Amharic sentence from a well-defined Interlingua representation with a native speaker acceptable feature.

**1.2 Motivation**

The motivations behind working on Amharic sentence generation are the following. Majorly, the absence of a generalized framework for Ethiopian languages to generate from semantic representation. This hinders Ethiopian languages from moving forward on various applications of NLP. On the other hand, sentence generation from knowledge based representation is gradually emerging and to our best knowledge there are no works done for Ethiopian languages to this point. This means that even though there are numerous amount of shared knowledge from multilingual environment worldwide, Ethiopian languages have no easy way to acquire such knowledge due to highly demands language independent representation like Interlingua for sharing of knowledge

and communicating in multilingual environment. Furthermore, the viability of Interlingua based NLG for Ethiopian languages which number many, but are closely knit in terms of kinship relations.

### **1.3 Statement of the problem**

Generation of sentences from Interlingua representations have been tested for different languages. Some researchers also tried to adopt works of other researchers. But, the results were not satisfying, even for less inflectional languages. The main reason behind the poor results of adopting is the dissimilar nature of natural languages. Morphological complexity is also the other major reason for improper fit with works done for other natural languages.

Thus, we need to develop Amharic sentence generator in order to generate text from Interlingua. The sentence that is generated from this system should be easily understandable. Implementing what has been done in other languages to our target language, Amharic, would not work effectively due to the following reasons. Primarily, Amharic is one of highly inflectional languages. Besides, its grammar creates complexity. Therefore, the research should come up with a solution for generating Amharic sentence from Interlingua representation.

### **1.4 Objectives**

#### **General Objective**

The general objective of this research is to develop Amharic sentence generator from Interlingua representation.

#### **Specific Objectives**

To achieve the general objective, the following specific objectives are identified:

- Review literature in the area of Interlingua based sentence generation.
- Collect Interlingua corpus.
- Prepare Interlingua to Amharic dictionary.
- Develop a model for representing Amharic grammar.
- Implement Amharic word generator and phrase constructor
- Develop a prototype for Amharic sentence generator.

- Test and evaluate the performance of the sentence generator.

## 1.5 Methodology

In order to achieve the above objectives, we will apply the following methods.

- **Literature Review:** Detail review and assessment will be made on works which are done in the area of sentence generation based on Interlingua and related issues to know the actual problem with this strategy. Morphological synthesizer, word ordering and grammatical rules will also be reviewed. After reviewing works done in this domain best approach will be applied.
- **Data Collection and Analysis:** Development of the system requires huge data source both from the Interlingua representation and Amharic words. For the Interlingua corpus, first we need to choose one standard representation and for Amharic we need to identify grammar rules and prepare a dictionary.
- **Tools:** -Different tools will be used to develop word synthesizer, phrase constructor and sentence generator.
- **Prototype Development:** -The prototype of the system will be developed to evaluate the performance of the proposed Amharic sentence generator. The evaluation could be done via the predefined standard in the area.

## 1.6 Scope and Limitations

The Amharic sentence generator will generate sentences only from one standard Interlingua representation. The representation shall be decided after the review of literature. The sentence generation will focus mainly on producing an acceptable sentence for the native speaker. In addition, the sentence produced by the generator will be ordered properly according to the rules of Amharic. Also, the translation of terms listed in the representation must be faithful to the original sense. Finally, the translated words will be furnished in a correct manner to easily transmit the concepts in the representation. However, this research work will be limited to simple and compound sentences and will not handle complex sentences.

## 1.7 Application of Results

The output of this research can be applied in various other NLP fields of researches. These can be:

- **Machine translation (MT)** - is transformation of a text written in one language into an equivalent text in another language [8]. This property of machine translation makes it multilingual. Always in machine translation at least two different languages should be involved. Interlingua approach is best for machine translation mainly because it requires fewer components in order to relate each source language to each target language. Even it takes fewer components to add a new language.
- **Dialogue Systems (DS)** - The target of DS is presenting information to users in an easy to understand format. Users ask the system specific information; once obtained, the system can show it through an answer in natural language. If the DS system could get the question in the Interlingua representation and provide the answers in an Interlingua representation, the Amharic generator shall address the rest of the problems, which includes sentential fluency and text planning.
- **Question Answering (QA)** – If the question can be changed to Interlingua representation the answer can be easily generated by the proposed Amharic generator.
- **Text Summarization (TS)** - Once the TS system capture the true sense of the text, it will paraphrase the concepts in a block text. Thus, our system can contribute by producing a grammatically correct sentence.

## 1.8 Organization of the Rest of the Thesis

The remaining part of this thesis is organized as follows. The details about the literature review carried out for this work is presented in Chapter Two. Reviews of related works that have been conducted on NLG, Interlingua and Universal Network Language (UNL) based MT are presented in Chapter Three. The details on the design and development of the Amharic sentence generator are given in Chapter Four. The Experiment on the implementation of the proposed work have been provided in Chapter Five. Finally, Chapter Six of this thesis discusses the conclusion and future work.

# Chapter Two: Literature Review

This Chapter contains the review of literature on the area of NLG. Different Interlingua representations are also discussed and more focus is given to Universal Network Language (UNL) representation. In addition, literatures about Amharic grammar is presented.

## 2.1 Natural Language Generation

For Reiter and Dale [2], NLG systems are computer software systems that start from some type of non-linguistic representation of information as input and rely on knowledge about language and the application domain to automatically produce reports, documents, explanations and other kind of texts. In short, NLG consists in the production of natural language texts from an abstract semantic knowledge representation. In this way, an NLG system takes into account abstract information (generally non-linguistic, unequivocal and well-structured information), as well as data about the communicative situation, in order to produce a text with a coherent structure and appropriate linguistic expressions. In the case of multilingual generation, the system is able to produce texts in several languages from the same abstract source, either in a simultaneous or sequential way.

There are six important activities that need to be carried out by an NLG system. These activities are content determination, discourse planning, sentence aggregation, lexicalization, referring expression generation and linguistic realization [9, 10].

The applications of NLG are varied: The Special Interest Group in Natural Language Generation (SIGGEN<sup>1</sup>) mentions report generation, machine translation, and explanations for knowledge-based systems; also, text summarization and multilingual and multimodal presentation of information. NLG systems can be used for preparing weather reports, visualizing the content of a database, generating automatic speech, etc., and highlight its application in the field of machine translation between natural languages and in the translation from a knowledge representation into numerous languages. NLG is a subtask of many applications: machine translation, human-

---

<sup>1</sup> We can find more information at the URL <https://www.aclweb.org/>. SIGGEN is the special interest group devoted to the study of NLG. [Last visited: 1-8-2016]

computer dialogue, summarization, report creation, automatic technical documentation, proof/decision explanation, customized instructions, item and event descriptions, question answering, tutorials, stories, and more [11].

NLG only makes sense when the data to be communicated is already present in an existing database (knowledgebase etc.), the users need the information presented as text and the volume of documentation justifies the expense of building an NLG system [12].

At the present time one can fairly easily build a single-purpose generator for any specific application, or with some difficulty adapt an existing sentence generator to the application, with acceptable results. However, one cannot yet build a general-purpose sentence generator or a non-toy text planner. Several significant problems remain without sufficiently general solutions. These includes lexical selection, sentence planning, discourse structure, domain modeling generation choice criteria [12].

Lexical selection is one of the most difficult problems in generation. At its simplest, this question involves selecting the most appropriate single word for a given unit of input. However, as soon as the semantic model approaches a realistic size, and as soon as the lexicon is large enough to permit alternative locutions, the problem becomes very complex [13]. In some situation, one might have to choose among the phrases *የየሃንስ መኪና*, *የየሃንስ የስፖርት መኪና*, *የእሱ የቤት መኪና*, *ቀዩ መኪና*, *ቀዩ መዝዳ* for referring to a certain car. The decision depends on what has already been said, what is referentially available from context, what is most salient, what stylistic effect the speaker wishes to produce, etc. A considerable amount of work has been devoted to this question, and solutions to various aspects of the problem have been suggested by Universal Networking Digital Language (UNDL) foundation [14].

Even assuming the text planning problem solved, a number of tasks remain before well-structured multi-sentence text can be generated. These tasks, required for planning the structure and content of each sentence, include: pronoun specification, theme signaling, focus signaling, content aggregation to remove unnecessary redundancies, the ordering of prepositional phrases, adjectives, and etc. While to the non-specialist these tasks may seem relatively unimportant, they can make the difference between a well-written and a poor-written text [15]. Theories of pronoun use, theme and focus selection and signaling, and content aggregation; implemented sentence planners with rules that perform these operations; testing in realistic domains is required.

It is generally agreed that the technology of NLG has evolved to a stage where it can feasibly be expected to be found in real world applied systems. A problem that remains outstanding, however, is that of the input to NLG applications: where should we get it from and what should it look like?

## **2.2 Interlingua Representation**

The motivation behind devising an Interlingua was the long-lived belief that while languages differ greatly in their “surface structures”, they all share a common “deep structure” [16]. The creation of universal representation in terms of deep structure, it enables natural languages with regularity and predictability they lack and in terms of surface structure, it enables natural languages to put concepts in a way that could be accepted by the corresponding language.

### **2.2.1 Interlingua Definition and Components**

An Interlingua is a system for representing the meanings and communicative intentions of language. It can be defined as a triple  $(S, N, L)$  where:  $S$  is a collection of representation symbols, often defined in an ontology, where each symbol denotes a particular aspect of meaning or intention (sometimes individually, and sometimes in concert with others according to specific rules of combination).  $N$  is a notation, within which symbols can be composed into meanings. The rules governing notational well-formedness reflect the compositional derivation of complex meaning out of ‘atomic’ symbols, an operation that is basic to the theory of the Interlingua.  $L$  is a lexicon, namely a collection of words of a human language such as English, in which each lexical element is associated directly or indirectly with one or more symbols from  $S$  [12].

In order to be capable of representing natural language content, an Interlingua should be, first, unambiguous; it should be more explicit even than the natural language it is representing [9]. Second, it should represent the full content of the input text; meaning its morphological, syntactic, semantic and even pragmatic characteristics [17]. Third, it should be universal, capable of representing the abstract meaning of any text, belonging to any domain or language. Fourth, an Interlingua should represent the content of the input alone and not be influenced by the formal representation of the content in the Source Language (SL) text [17]. Fifth and finally, the Interlingua should be independent of both the SL and the Target Language (TL); [18].

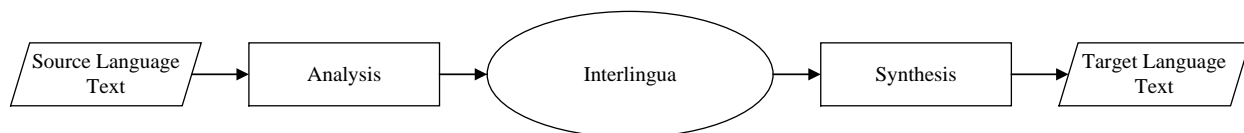
The advantages of using such an approach include economy, modularity, localization, back translation possibility and potential uses in other NLP-related areas such as Language Analysis,

Language Synthesis, Cross-Lingual Information Retrieval (CLIR), cross-language summarization, MT and multilingual QA [9, 17, 19, 20].

All of the previous research and projects exhibit intriguing approaches to defining an abstract language-independent format for knowledge representation. However, they vary in the degree of language-independency, the complexity through which they achieve such a representation and in their capabilities.

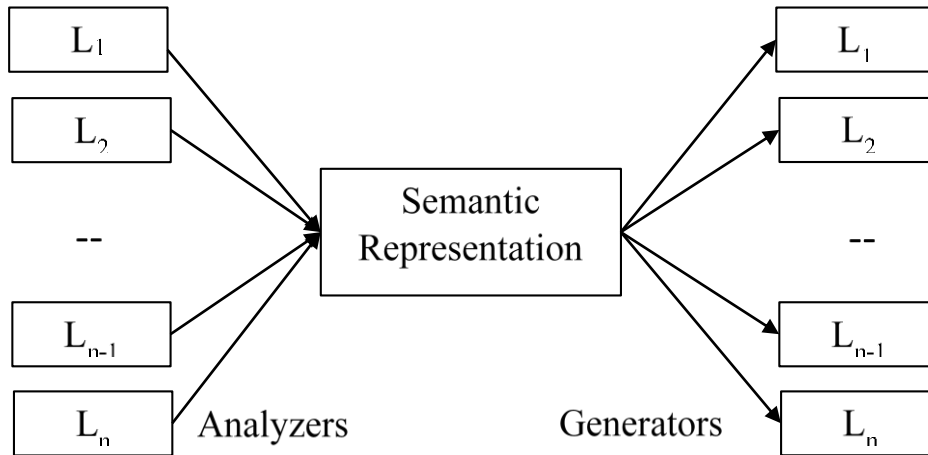
### 2.2.2 Interlinguas in Machine Translation

A typical Interlingua system is illustrated schematically in Figure 2.1 [12, 16]. Each language requires an analyzer and a synthesizer. The analyzer takes as input a source language sentence and produces as output an Interlingua representation of the meaning. The synthesizer takes an Interlingua representation of meaning as input and produces one or more sentences with that meaning. In theory, it is not necessary to consider the properties of another language during the analysis of the source language or generation of the target language. To translate from language L1 to L2, L1's analyzer produces an Interlingua representation and L2's synthesizer generates an L2 sentence with the same meaning.



*Figure 2.1: Interlingua MT System Architecture*

The system involves extensive use of source and target grammar rules during this process. Interlingua approach is best suited for multilingual machine translation, as it requires only two components for each language: one for conversion from source language to Interlingua and other for Interlingua to target language. As such, in this approach we require  $2n$  components as shown in Figure 2.2 [18]. The meaning based representation known as Interlingua makes this approach suitable for information retrieval applications as well.



*Figure 2.2: Interlingua-based Translation among n Languages*

There are few problems with Interlingua approach. The analysis of source text requires a deep semantic analysis, which sometimes results into the loss of information. Another problem is the creation of an adequate Interlingua, because it should be both abstract and independent of the source and target languages, which makes this task difficult in case of large multilingual machine translation systems involving the languages with wider differences.

There are some well-known interlingua-based systems; the most commonly used interlingua-based machine translation projects. Below we illustrate using representative examples of Interlingua representations used by developers of Interlingua MT systems.

### **DLT**

Distributed Language Translation (DLT) is a research project in Utrecht, The Netherlands [21]. Preliminary research in the project began as early as 1979. In 1984, DLT entered a six-year project to build MT system capable of translating from simplified English into French. However, in 1990, the DLT pilot project came to an end after receiving a fair amount of publicity.

DLT is an interactive system developed to operate over computer networks. Translation is distributed between two independent terminals; one for the analysis and another for generation. In the DLT system, the intermediate representation (the Interlingua) is a ready-made logical language with supposedly standardized rules for vocabulary and structures, i.e., Esperanto.

Semantic and Pragmatic knowledge constitute the language-independent component of the system and is completely handled in the intermediate stages of forming the Esperanto representation. Language specific information, on the other hand, is purely syntactic and is developed for a specific pair of languages, in one translation direction only [21].

## **UNITRAN**

UNIversal TRANslator (UNITRAN) is a translation system developed at Massachusetts Institute of Technology [22]. The system operates bi-directionally between Spanish and English. However, other languages may be added by setting the parameters that fit them. The UNITRAN system comprises two main components between which processing tasks are divided; the syntactic component and the lexical-semantic component. The syntactic component is based on the Government and Binding theory. It is responsible for handling the language-specific syntactic differences by accepting and producing grammatically correct sentences. The syntactic component is composed of a set of parameters associated with universal principles. These parameters are built-in in the analyzer and generator to be set according to the values of the language being processed. Thus, the analyzer and generator used are the same for all languages. The lexical-semantic component, on the other hand, is based on the Lexical Conceptual Structure (LCS) theory. It contains the information necessary to provide a conceptual form (the LCS) to underlie the source language sentence, and to match it to the appropriate target-language lexical items.

## **KANT**

Knowledge-based, Accurate Natural-Language Translation (KANT) system has been developed at Carnegie Mellon University in Pennsylvania, USA in 1989 [22]. It has been used in translating English technical documents into French, Spanish and German. The addition of more target languages such as Portuguese, Italian, Russian, Chinese and Turkish is under research. The KANT prototype has also been used in generating Japanese and German. The system codes for analysis and generation are language-independent whereas the specific knowledge required to process a certain language (grammars and lexicons) is developed separately for each language.

## **UNL**

UNL project was launched in 1996 at the Institute of Advanced Studies of the United Nations University, Tokyo, Japan [14, 20]. In January 2001, the United Nations University set up an

autonomous non-profit organization in Geneva, Switzerland, to be responsible for the development and management of UNL; the UNDL Foundation. In addition, 17 language centers all over the world are working on the development of the UNL resources necessary for incorporating their native language into the UNL program.

UNL is not intended to be an auxiliary language such as Esperanto, Interlingua, Ido or others. It is rather a formal artificial language that replicates the functions of natural language in communication, but is, nevertheless, designed for computers rather than humans. People should use UNL in “communication” in the same subtle manner they do with other procedural languages such as HTML [22, 23].

Although MT is one of the possible and more obvious and promising uses of UNL, it is not the only area in NLP where UNL can prove useful. As it offers a complete understanding of natural language content, UNL can serve areas such as summarization and text simplification. Moreover, by providing a language-neutral representation of meaning, UNL can dramatically improve our ability to search for and find information, thus, helping areas such as information retrieval and others.

From the four Interlingua based systems, UNL has succeeded in mending some of their imperfections that impeded reaching the ultimate goal of bringing down the language barriers [22, 23]. Hence, a large section of processing stages in most interlingua-based MT system is devoted to disambiguating the input to form the interlingua, and in some cases, disambiguating the intermediate representation to derive the output (as in the DLT system). In other cases, the system enforces very strict limitations on input language and imposes precise semantic restrictions on the abstract concepts to avoid prolonged processing (as the case in KANT). As a last resort, some systems turn to interactive communication with the user(s) such as DLT. UNL has largely avoided such intricate procedures by using a quite unambiguous intermediate representation. In UNL, a word can never have more than one conceptual representation. They are clearly distinguished by the ID number that represents their exact contextual meaning [22].

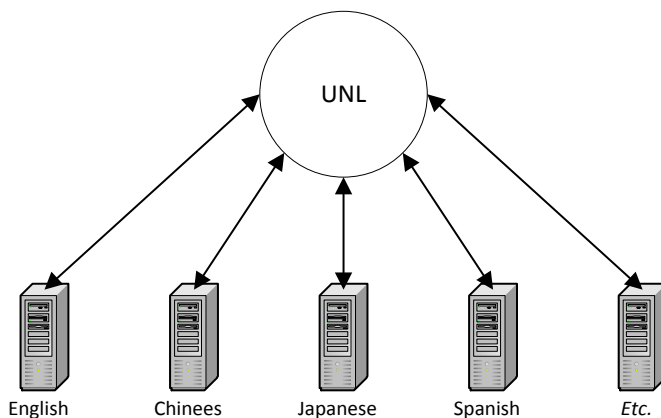
Moreover, most interlingua-based MT systems miss one aspect of meaning or another such as UNITRAN that does not incorporate the notion of grammatical aspect. UNL, on the other hand, tries to convey all aspects of meaning in its intermediate representation; semantic, syntactic, pragmatic, subjectivity, format, etc. UNL is not simply an intermediate representation; it is a full-

scale language for machines. This means that its uses go beyond the task of translation as mentioned earlier. Besides, it can represent any imaginable concept because, unlike a system such as UNITRAN which builds concepts from a limited set of primitives, it makes use of dozens of semantic Relations and Attributes to exactly convey the intended meaning [22].

### 2.2.3 UNL Expression

UNL is a language that enables computers to process information across different languages. It is used to replicate the functions of natural language used by humans for communication. It has the capability to represent information and knowledge provided by natural language. The motivation behind the development of UNL is to propose an Interlingua representation such that semantically equivalent sentences of all languages have the same representation [14, 20]. From the very beginning, researchers from all over the world have been working on creation of linguistic resources and software for UNL systems.

The EnConverter and DeConverter are the core software in a UNL system. The process of converting a source language (natural language) expression into the UNL expression is referred to as EnConversion and the process of converting UNL expressions into a target language representation is called DeConversion. This two software perform their functions on the basis of a set of grammar rules and a word dictionary of target language. UNL Language Servers can be connected to the Internet to carry out the conversions between natural languages and UNL expressions. Each Language Server will thus contain an EnConverter and a DeConverter of a language as shown in Figure 2.3 [14, 24].



*Figure 2.3: A UNL System*

## UNL Format for Information Representation

UNL represents the information in three different types of semantic units, namely, Universal Words (UWs), Relations and Attributes. UNL represents information sentence by sentence [14]. Each sentence is converted into a hyper-graph (also known as UNL graph) having concepts represented as nodes and relations as directed arcs. The concepts are represented by UWs and UNL relations are used to specify the role of each word in a sentence [20]. The subjective meanings intended by the author are expressed through UNL attributes. UNDL has formally defined the specifications of UNL [24].

### Universal Words

Universal Words form the vocabulary of UNL. These words correspond to the nodes that are interlinked by relations or modified by attributes in a UNL graph. The concepts of UWs are divided into four categories, namely: nominal concepts, verbal concepts, adjective concepts and adverbial concepts. A UW is a character string (an English language word) followed by a list of constraints [24]. The syntax used for describing a UW is:

```
<UW> ::= <headword> [<constraint list>]
```

Here, the headword is an English word or a phrase or a sentence that is interpreted as a label for a set of concepts. This is also called a basic UW. The constraint list is used to describe the sense of word. This list contains the UNL relations that represent the objective knowledge of concept in headword. The constraint list is also used to restrict the range of the concepts that an English expression represents and thus also helps in disambiguation.

The UWs are divided into four types: Basic UWs; Restricted UWs; Extra UWs and Temporary UWs. Basic UWs are bare headwords with no constraint list, for example, '*water*', '*go*', '*run*' etc. A UW with constraint list is known as Restricted UW. For example, the basic UW '*run*', with no constraint list, denotes the concepts of 'management of an organization by a person' (e.g., 'He is running the organization very well'), 'driving of vehicle' (e.g., 'He runs the car very fast'), 'move at a speed faster than walking' (e.g., 'He runs very fast in the race'), 'accumulation of bill' (e.g., 'Your bill runs from January 1, 2011 to December 31, 2011'), 'movement of finger on keyboard' (e.g., 'She ran her fingers on the keyboard'), 'taking risk by a person' (e.g., 'Owing to his habitual lateness he ran the danger of being fired'), 'short visit' (e.g., 'He runs up to Kebena'),

and so on. The UW 'run' is restricted with constraint list to represent these concepts unambiguously as shown in Listing 2.1.

```
run (agt>human, equ>manage, obj>organization)
run (agt>human, equ>drive, obj>vehicle)
run (agt>human)
run (agt>human, equ>accumulate, obj>bill)
run (agt>human, obj>risk)
run (agt>human, opl>keyboard, obj>finger)
run (agt>human, pur>short visit)
```

*Listing 2.1: UW 'run' Restricted with Constraint List*

Here, 'agt', 'equ', 'obj', 'opl' and 'pur' are UNL relations. Extra UWs are special type of restricted UWs that are used to represent the concepts that are not available in English language [20]. These words are represented as headwords using English characters and constraint list gives idea about the nature of the concept. For example, *samba(icl>dance)* is “a kind of dance”, *'eskista(icl>dance)'*, *'soufflé(icl>food)'* is “a kind of food”, *'ikebana(icl>flower arrangement)'* is “a kind of flower arrangement” for the meaning of “something you do with flowers”, etc.

Temporary UWs may not have a definition. For example, a number or an e-mail or URL need not be necessarily defined. These can appear in a UNL document and are treated as temporary UWs. The words from other languages are linked to these UWs and are assigned syntactic and semantic attributes. This forms the core of the lexicon building activity. In the UNL specifications proposed in 2010, a simple UW is considered as an integer that can also be represented as a unique character string for better readability [21]. This character string is split into two different parts: a root and a suffix. The root can be a word, an expression, a phrase or even an entire sentence in a language which can be interpreted as a label for a concept. The suffix is used to disambiguate the root. The UWs are considered equivalent to the sets of synonyms of a given language, similar to the concept of synsets in a WordNet. The UWs have been proposed in such a manner that language groups can create their own language specific UNL dictionaries automatically with the use of their

corresponding language WordNet. The UWs are organized in a hierarchy (the UNL ontology), which are defined in the UNL knowledge-base and exemplified in the UNL example-base (lexical databases for UNL) provided by UNDL center [14].

### UNL Relations

Relations are basic building blocks of UNL sentences. UNL expressions are formed using binary relations and a binary relation includes a UNL relation and two UWs. The relations between UWs have different labels according to the different roles they play. There are a number of factors that should be considered in choosing a relation [19, 24]. A UNL binary relation is of the form  $rel(arg1, arg2)$ . Here,  $rel$  represents the label of UNL relation and its first argument  $arg1$  is UW1 and second argument  $arg2$  is UW2. The UW1 acts as the parent of the relation whereas UW2 acts as the child of the relation. There are 46 relations in all that are defined for UNL [14, 24]. The description of these relations is given in Annex A.

### UNL Attributes

Attributes are used to describe subjective information in a sentence. These attributes exhibit the point of view of the speaker in a given sentence [19]. There are 87 attributes (which can be augmented with the user defined ones) to express the semantic content of a sentence. UNL attributes are divided into eight groups. Descriptions of these groups with corresponding UNL attributes are given in Table 2.1 [24].

*Table 2.1: Description of UNL Attributes*

Concept	Attributed as
Logicality of UW	@transitive, @symmetric, @identifiable, @disjointed
Time with respect to the speaker	@past, @present, @future
Speaker's view on aspects of event	@begin, @complete, @continue, @custom, @end, @experience, @progress, @repeat, @state @just, @soon, @yet
Speaker's view of reference to concepts	@generic, @def, @indef, @not, @ordinal

Concept	Attributed as
Speaker's emphasis, focus and topic	@contrast, @emphasis, @entry, @qfocus, @theme, @title, @topic
Speaker's attitudes	@affirmative, @confirmation, @exclamation, @humility, @imperative, @interrogative, @invitation, @polite, @request, @respect, @vocative
Speaker's feelings and judgments	Attributes to represent ability: @ability Attributes to represent beneficially: @get-benefit, @give-benefit Attributes to represent conclusion: @conclusion, @consequence Attributes to represent condition: @sufficient Attributes to represent consent/dissent: @consent, @dissent, @grant, @grant-not Attributes to represent expectation: @although, @discontented, @expectation, @wish Attributes to represent intention: @insistence, @intention, @want, @will, @need, @obligation, @obligation-not, @should, @unavoidable Attributes to represent possibility: @certain, @inevitable, @may, @possible, @probable, @rare, @unreal Attributes to represent emotion: @admire, @blame, @contempt, @regret, @surprised, @troublesome
Convention description	@passive, @pl, @angle_bracket, @brace, @double_parenthesis, @double_quote, @parenthesis, @single_quote, @square_bracket

Attributes are thus used to include information about time and aspect of the event, modality of the predication, reference of the entities mentioned, number and/or gender, etc. For example, in the sentence, 'The boy eats potatoes in the kitchen', attributes are needed to express plurality in the object ('potato', i.e., '@pl'), to indicate definite reference to agent ('boy', i.e., '@def'), to indicate definite reference to place ('kitchen', i.e., '@def') and to denote the head of expression ('eat', i.e., '@entry') [14].

### UNL Sentence

UNL sentence is the basic unit of representation inside the UNL framework. It consists of nodes (UWs) interlinked with binary semantic relations and modified by attributes. There are two different ways of representing UNL sentences: the table format and the list format. In the table format, UWs and relations constitute a single structure, while in list format, UWs and relations are

represented separately [24]. The UNL sentences are commonly represented in the table format. The syntax of table format is given in Table 2.2 [24].

*Table 2.2: Syntax of UNL Sentence in Table Format*

<b>&lt;UNL sentence&gt;</b>	<b>::=</b> <list of relations>
<b>&lt;list of relations&gt;</b>	<b>::=</b> <binary relation>[<binary relation>...]
<b>&lt;binary relation&gt;</b>	<b>::=</b> <relation>["<Scope-ID>] "(" <source node>, <target node>")"
<b>&lt;source node&gt;</b>	<b>::=</b> <UW+attributes>
<b>&lt;target node&gt;</b>	<b>::=</b> <UW+attributes>
<b>&lt;UW+attributes&gt;</b>	<b>::=</b> <UW>{"<Scope-ID> [<attribute list>] ":" <UW-ID>

Here, '<' and '>' indicates a non-terminal symbol; '{' and '}' indicate a range; '[' and ']' indicate an omissible part; '...' indicates more than 0 times repetition of the front part; ':=' indicates that left part can be replaced by the right part and predefined delimiters are enclosed within "". The syntax of list format is given in Table 2.3 [24].

*Table 2.3: Syntax of UNL Sentence in List Format*

<b>&lt;UNL sentence&gt;</b>	<b>::=</b> "[W]" <list of UWs> "[/W]" [ "[R]" <list of relations> "[/R]" ]
<b>&lt;UW+attributes&gt;</b>	<b>::=</b> <UW+attributes> [<UW+attributes>...]
<b>&lt;UW+attributes&gt;</b>	<b>::=</b> <UW>{"<Scope-ID> [<attribute list>] ":" <UW-ID>
<b>&lt;list of relations&gt;</b>	<b>::=</b> <binary relation>[<binary relation>...]
<b>&lt;binary relation&gt;</b>	<b>::=</b> <source node><relation["<Scope-ID>]<target node>
<b>&lt;source node&gt;</b>	<b>::=</b> <UW-ID>

```
<target node> ::= <UW-ID>
```

These formats are illustrated below with the help of an example sentence.

The boy eats potatoes in the kitchen.

The UNL sentence in table format of the above sentence is depicted in Listing 2.2.

```
{unl}
agt(eat(icl>do).@entry, boy(icl>male person).@def)
obj(eat(icl>do).@entry, potato(icl>food).@pl )
plc(eat(icl>do).@entry, kitchen(icl>facilities).@def)
{/unl}
```

*Listing 2.2: UNL Expression Table Format Example*

From Listing 2.2, 'agt', 'obj' and 'plc' are the relation labels, 'eat(icl>do)', 'boy(icl>male person)', 'potato(icl>food)' and 'kitchen(icl>facilities)' are the UWs, '@entry' is used to indicate head of whole expression, '@def' is used to indicate definite reference to agent and place and '@pl' is used to indicate plurality of object.

The UNL sentence in list format form for the example sentence given above is:

```
{unl}

[W]

boy(icl>male person).@def :01
eat(icl>do).@entry :02
potato(icl>food).@pl :03
kitchen(icl>facilities).@def :04

[/W]

[R]

02agt01
02obj03
02plc04

[/R]

{/unl}
```

*Listing 2.3: UNL Expression List Format Example*

The list format of UNL sentence representation given in Listing 2.3, suggests that UWs are sequenced with UW-ID from '01' to '04' under *[W]* and *[/W]* tags. The list of relations in the form of source node UW-ID, relation, and target node UW-ID is listed under *[R]* and *[/R]* tags [24].

As indicated earlier, a UNL sentence represented either a table format or list format can also be depicted in the form of a graph called as UNL graph. The UNL graph of

Listing 2.2 and/or Listing 2.3 is depicted in Figure 2.4.

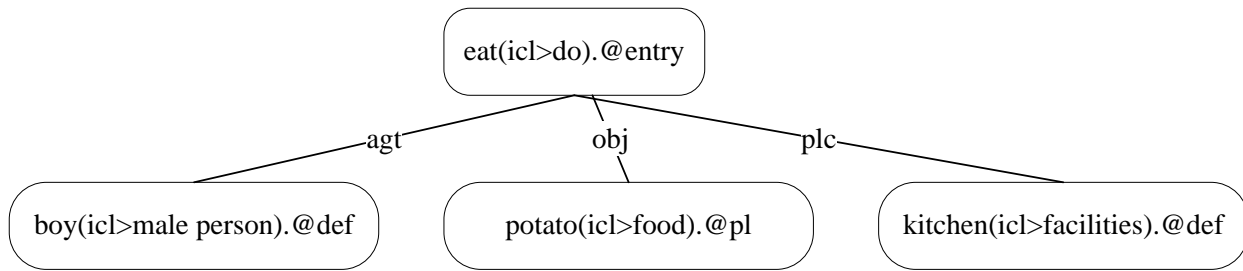


Figure 2.4: UNL Graph for Example Sentence Given In Listing 2.2 and/or Listing 2.3

According to the UNDL standard the data format of a word dictionary entry is as mentioned below:

[HW] {ID} "UW" (ATTR, ... ) <FLG, FRE, PRI>;

An entry must end with a semicolon. <FLG, FRE, PRI> can be omitted. The description of each dictionary entries are depicted in Table 2.4 [24].

Table 2.4: Description of Word Dictionary Entries

HW	The headword of a native language <i>i.e.</i> , equivalent to Amharic lexeme
ID	The identifier of the headword. Leave this field empty if it is not necessary to fill. This field is useful for the Co-occurrence Relation Reference. The detail is shown in <Co-occurrence Dictionary development and directions for use>
UW	The Universal Word of UNL. This field can be empty if a word does not need an UW.
ATTR	The Grammatical Attributes of the headword. Such as (N, FEEL), a separator”,” is necessary between each two attributes.
FLG	The Language Flag of one character of English in ASCII code. This field can be omitted together with FRE and PRI. If it is omitted, the system will give the value of “.” to it.
FRE	The Frequency of the headword. This field can be omitted together with FLG and PRI. If it is omitted, the system will give the value of 0 to it.
PRI	The Priority of the headword. This field can be omitted together with FLG and FRE. If it is omitted, the system will give the value of 0 to it. The value range of the PRI is between 0 and 255.

## 2.3 Amharic Grammar

Amharic /አማርኛ/ is a member of the Semitic branch of the Afro-Asiatic language family. It is a Semitic language spoken in Ethiopia. It is the official working language of the Federal Democratic Republic of Ethiopia which is also the second most-spoken Semitic language in the world, after Arabic. Ethiopia is the second most populous country of Africa having more than 73.75 million inhabitants according to 2007 Census. Based on a projection of annual growth rate of 2.6 percent, Ethiopia's population up to July 2016 is more than 90 million and Amharic is spoken as a first language by more than 27 million people of the country [30].

Nouns in Amharic are inflected for number (singular/plural), species (definite/indefinite) and case (nominative/accusative/genitive/dative) [33]. Unlike these variable features, the nouns exhibit an inherent behavior towards gender, that is, a given Amharic noun is either masculine or feminine. These inflections are usually achieved through affixation. Suffix affixations are predominant while there are also a good number of prefixes. The above order shows the way in which these nominal affixes appear. We show this by giving an example “*the houses*” (definite, a plural noun phrase used as a direct object) from the sentence “*I sold the houses.*” and show the Amharic counterpart as follows:

the houses - ቤቶቹን

From the above noun, suffixes (“-ኸኸ”, “-ኡ” and “-ን”) are added to the head noun “ቤት”, which means house, to mark inflections for number (Pl), species (Def) and case (Acc) to get the required form “the houses -ቤቶቹን”.

Markos Kassa Gobena [35] defines nouns inflect in number (singular or plural), species (definite or indefinite) and case (Nominative, Accusative, Genitive or Dative) while a given noun has an inherent gender.

Pl Def Acc: ቤቶቹን

From the above example, we can see that the noun ‘ቤት’ is inflected for plural number by adding the suffix ‘-ኸኸ’, definite species by adding ‘-ኡ’ and accusative case by adding ‘-ን’.

The pronouns in Amharic can be put into three persons as in English, but there are some unique features such as the second person 'you' which may take different agreements when referring 'plural', 'respected (politeness)', 'singular-female' or 'singular-male' nouns as shown below [35].

You (Masc. Sg.):	አንተ
You (Fem. Sg.):	አንቺ
You (Pl.):	እናንተ
You (Politeness):	እርስዎ

*Listing 2.4: Amharic Pronoun Example*

Adjectives come before nouns in a sentence to modify them [33]. In a sentence “አንተ በጣም ጎበዝ ተማሪ ነህ” – “you are a very clever student” there are two adjectives “በጣም” - *very* and “ጎበዝ” – *clever*. “በጣም” - *very* modifies “ጎበዝ”- *clever*, and “ጎበዝ” modifies the noun “ተማሪ” – student.

Amharic verbs are the most complex category of words. They are created generally from consonantal radicals, which are inflected by a process of merging with vocalic components based on various patterns [33, 34]. The majority of roots, like other Semitic languages, have three radicals. However, there are also a significant number of verbs that are multi-radical and bi-radical. A verb takes various forms depending on the tense-aspect-mood, voice and root structure while inflecting for person gender and number. This is done through the addition of affixes at both ends and even between the roots of a stem [35].

Markos Kassa Gobena [35] defines the different Amharic radicals by using record types that help to represent words as data structure as follows:

```
Root2: Type = {C1, C2: Str};
Root3: Type = Root2 ** {C3: Str};
Root4: Type = Root3 ** {C4: Str};
Pattern2: Type = {C1, C1C2, C2: Str};
Pattern3: Type = {C1, C1C2, C2C3, C3: Str};
Pattern4: Type = {C1, C1C2, C2C3, C3C4, C4: Str};
```

Where C1, C2, C3 and C4 stand for the consonantal radicals while, for instance in Pattern2 C1, C1C2 and C3 represent the head, the middle and the tail of the pattern. Hence, Root2 is bi-

radical while Root3 has an additional one consonant making it a tri-radical and similarly Root4 is quadri-radical.

When root word is given to the system, it associates every consonant in the input string with a variable. These variables are then coded into patterns given a root and a pattern. The final output is just a concatenation of strings. Consider the following example patterns:

C1aC2aC3a = {C1 = ""; C1C2="አ"; C2C3 ="አ"; C3="አ"};  
 meC1C2aC3 = {C1 ="መ"; C1C2=""; C2C3 ="አ"; C3=""};

For instance, when the root ስብር is applied to the first pattern C1aC2aC3a, it forms the perfect stem ሰበረ while when applied to the second pattern it forms an Amharic infinitive መስብር.

According to Markos Kassa Gobena [35], Verb lexemes are inflected for person, number, gender, voice and form. Since the system considers the morphology generation of a word, the agreement with other nouns are not considered.

The numerals in Amharic can assume a cardinal or ordinal form. The ordinals are all the times achieved by adding the prefix -አኛ on the ordinals.

Cardinal → ሁለት

Ordinal → ሁለት+አኛ → ሁለተኛ

The prepositions appear as simple prepositions that are standalone or as separate entities coming both at pre-and post-positions [33, 34].

to you ለ - አንተ (pre)

on - you አንተ - ላይ (post)

with - you ከ - አንተ - ጋር (both pre-and post)

### 2.3.1 Phrase Construction

A phrase is a structure in a language that is constructed from one or more words in the language. In Amharic, phrases are categorized into five, namely noun phrase, verb phrase, adjectival phrase, adverbial phrase and prepositional phrase [31, 33]. Each phrase type can be categorized into “simple” where only one word class is represented and “complex” where more than one word classes are represented.

### **Noun Phrases**

A Noun Phrase (NP) is a syntactic unit in which the head (H) is a noun or a pronoun. It can be simple or complex. The simplest NP consists of a single noun (e.g. Abebe) or pronoun such as እሱ, እሷ, እነሱ, etc. A complex NP can consist of a noun (called head) and other constituents (like complements, specifies, adverbial and adjectival modifiers) that modify the head from different aspects [33]. For example, in the NP ያ ያለፈው ሳምንት ቆንጆ መኪና, “ያ” is a specifier, ያለፈው ሳምንት is an adverbial modifier, ቆንጆ is an adjectival modifier specifying the material from which the object named by the መኪና (head) is made of.

### **Verb Phrases**

A Verb Phrase (VP) is composed of a verb as a head and other constituents such as complements, modifiers and specifiers. For example, in the VP ወደ ስራ ሄደች, ወደ ስራ is Prepositional Phrase (PP) modifying the verb ሄደች from the place point of view [33].

### **Adjectival Phrases**

The construction of Amharic Adjectival Phrase (AdjP) is similar to that of a NP and a VP. It can be composed of an adjective (head), and other constituents such as complements, modifiers and specifiers. For example, in the AdjP ያ በጣም የሚያምር, ያ is a specifier, በጣም is a modifier modifying the head of the AdjP, የሚያምር [33].

### **Prepositional Phrases**

Prepositional Phrase (PP) is constructed from a preposition head and other constituents such as nouns, NP, verbs, VP, etc. In the PP እንደ እንሰሳ በዱር, for instance, እንደ and በ are prepositions which are combined with the nouns እንሰሳ and ዱር, respectively to form their PPs. The two PPs, in turn, combine to result in the bigger PP that is provided in the example [33].

### **Adverbial Phrases**

An Adverbial Phrase (AdvP) is constructed from one or more adverbs in the language. In the example, ከፋኛ ታማሎች, ከፋኛ (head) is the only adverb that formed the AdvP [33].

## **2.3.2 Sentence Construction**

Sentence construction involves words and phrases of a language ordered according to the rules that govern the language to make sense. Word order has different form in different language. This

word order includes the general structure of the sentence, the modifier and modified word order (Adjective and noun or adverb and verb) and the like. Amharic has Subject Object Verb (SOV). Sometimes, in informal texts, it may occur as Object Subject Verb (OSV). However, this OSV order cannot be used in formal Amharic texts. Depending on the natural language used, the words in a sentence should be in their correct order. For instance, let's consider a simple Amharic sentence “አበበ በሶ በላ”. “አበበ” is a noun which is the subject of a sentence, “በሶ” is also a noun which is the object of a sentence and “በላ” is the verb. Moreover, the verb agrees with subject and object of the sentence [31, 33].

The word order also includes the adjective and noun order, adverb and verb order, verb and sentence end and the other Part of Speech tags in a sentence. For example, adjectives should appear before the noun that it modifies and adverbs also before the verb it modifies.

**Agreement with verb**

The word order in Amharic clauses is generally SOV. Verbs agree with their subjects in number, gender and person and objects precede verbs within the verb phrase . The subject of the verb is marked by suffixes or prefixes on the root verb. Object agreements are most often indicated by an object pronoun suffix because, unlike the markers of subject agreement, they do not vary significantly with the tense, aspect, mood of the verb [31, 33].

The subject and the verb in a sentence must agree in number, person and gender [33]. The following example illustrates subject-verb disagreement:

1. አነርሱ ቤታቸው ተቃጠለባት።
2. እኔ በጣም ስለቸኮልኩ እየሮጥኩ ወደቤት ሄደኝ።
3. ዘፋኙ ኢትዮጵያዊ ነች።

The first sentence in the example above shows number mismatch in Amharic language. “አነርሱ” is a subject in a plural form, whereas “ተቃጠለባት” is a verb in singular subject form. The second sentence encounter person mismatch. “እኔ” is a first person singular and “ሄደኝ” is a verb in third person singular subject form. In the third sentence, “ዘፋኙ” is a masculine whereas “ነች” is a verb in feminine subject.

Like the subject- verb agreement, object and verb must also agree in number, person and gender.

1. የአበበ ደብተሮች ተሸጠ።

2. እኔ አንቺን እወዳታለሁ።
3. ይትባረክ ቦጊቱን ወሰደው።

The first sentence in the example above shows number mismatch between object and verb. “ደብተሮች” is an object in a plural form, whereas “ተሸጠ” is a verb in singular form. The second sentence encounter person mismatch. “አንቺን” is a second person singular object and “እወዳታለሁ” is a verb in third person singular. In the third sentence, “ቦጊቱን” is an object with feminine gender whereas “ወሰደው” is verb in a masculine gender for the object.

### **Adjective and Noun Agreement**

Modifiers are words or phrases that are used in a sentence to elaborate something. Modifiers can be adjectives which modify noun or adverbs that modify verbs. Amharic adjectives appear before nouns that they modify. Amharic adjective should agree with the noun it modifies in number and gender [31, 33, 36].

Adjective is used in a sentence to give a clear picture of the noun that it modifies. It is simply a word that tells more about the noun. Amharic language has adjectives to modify nouns. These adjectives may mark number (plural and singular) and gender (feminine and masculine) of the noun they modify. If an adjective marks the number and gender of the noun, the marker should agree with the number and gender of the noun [31, 36].

For example:

“ትልቋ ላም” is a noun phrase that contains adjective “ትልቋ” and noun “ላም”. The noun “ላም” has a feminine character and the adjective “ትልቋ” has a feminine gender marker in it. If this phrase is rewritten like “ትልቋ ላም”, then it would be grammatically incorrect as the gender marker (which is masculine marker) in the adjective does not match with the gender of the noun. Therefore, to make the correction either the noun should be changed to masculine gender or the gender marker in adjective should be changed.

“ትልቅ ላሞች” is grammatically incorrect as it contains a singular marker in the adjective and the noun it modifies is a plural noun. This can be corrected by adjusting number marker either on the adjective or the noun. “ትልልቅ ላሞች” is grammatically correct with plural noun and plural marker in the adjective. Both the adjective and the noun it modifies have a plural marker and agree in number. “ትልቅ ላም” is also correct with singular marker in the noun and adjective.

## Adverb and Verb Agreement

Another type of modifier is an adverb. An Amharic adverb usually modifies the verb that comes next to it. Adverbs are few in number and can be found in primitive (as separate words that appear by themselves) and compound form (as combination of two words appearing as a word or detached words). As in many languages Amharic adverbs are classified into subclasses such as adverbs of time, place, circumstance, etc. Time adverbs describe the time at which an event takes place. These adverbs may show a specific time at which a given action takes place or its duration. Amharic verbs indicate the time at which an action takes place in relation with the adverbs. In Amharic (like many other languages), there are a set of forms taken by a verb to indicate time. These forms (tenses) can be grouped into seven [31, 36, 37].

In the sentence “እህቴ በሚቀጥለው ሳምንት አገባች።”, ‘በሚቀጥለው ሳምንት’ which is in the future and the verb ‘አገባች’ expresses the action that is already performed in the past. Therefore, it should be corrected as either ‘እህቴ በሚቀጥለው ሳምንት ታገባለች።’ or ‘እህቴ ባለፈው ሳምንት አገባች።’. In these two sentences, the adverb and verb agree in time.

## 2.4 Morphological Generator

A morphological analyzer is used to analyze the internal structure of the words of a language. On the other hand, a morphological generator does exactly the reverse of it, i.e., given a root word and grammatical information, a morphological generator will generate the particular word form of that root word [38]. The role of morphology is very significant in the field of NLP, as seen in applications like MT, QA system, IR, Spell checker, lexicography, etc. [38, 39].

Morphological synthesis or generation is a process of returning one or more surface forms from a sequence of morpheme glosses. There are many language dependent and independent approaches used for developing morphological generator [38]. These approaches can also be classified generally into corpus based, rule based and algorithmic based. In corpus based approach, a large sized well generated corpus is used for training with a machine learning algorithm. The performance of the system will depend on the feature and size of the corpus. The disadvantage is that corpus creation is a time-consuming process.

On the other hand, rule based approaches are based on a set of rules and a dictionary that contains roots and morphemes. In rule based approaches every rule depends on the previous rule. So if one

rule fails, it will affect the entire rules that follow it. When a root word is given as an input to the morphological synthesizer and if the corresponding morphemes are missing in the dictionary then the rule based system fails. Literature shows that there are a number of successful morphological generator development for languages like English, Chinese, Arabic and European languages using these approaches [39].

## 2.5 Summary

In this Chapter, we discussed about NLG and the major components involved. We also discussed about sentence generators and the challenges faced during development of sentence generator. Then, we tried to define and identify the advantages of Interlingua followed by discussing the representative examples of Interlinguas. Considering some literature, we also tried to choose which Interlingua representation to use and we found UNL to be eligible for our work. Even though there are few literatures written for Amharic grammar, we have reviewed some of them.

In multi-lingual environment, it is very important to have sentence generator systems for large number of languages. In order to have a multi-lingual NLG system, it would require a separate system for every language, *i.e.*,  $n*(n-1)$  language systems will be required for  $n$  number of languages. As such, there is a need to develop an internationally standardized system in which fewer components are needed for a multi-lingual NLG. An Interlingua based approach is one solution for this problem because it requires few components to learn generating from each source representations to each target language. When adding a new language like Amharic, just being able to generate from the Interlingua to Amharic is enough because all the concepts are already represented in the Interlingua. Besides, UNL is an internationally standardized Interlingua that has the potential to bridge the language barriers across all the languages of the world.

## Chapter Three: Related Work

This Chapter presents a review of related works that have been done in the area of NLG using UNL. We also discussed what has been done on sentence generation for different languages. Finally, those MT systems developed using UNL are also reviewed

### 3.1 UNL Based Sentence Generation

Singh *et al.* [4] proposed about generating Hindi sentences from an Interlingua representation called UNL. This system is a step towards machine translation involving Hindi as the target language. In addition, they consider the requirement of keeping the generated language natural and native speaker acceptability as a challenge. They have tried to adapt the universal DeConverter tool Deco, provided by the UNDL foundation. The source code for Deco is not available and its rule-format is complex since it aims to be Turing complete. They also tested Manati, their proposed solution which is simpler than Deco but it is still a complex framework because it also is a universal DeConverter. This motivated the researcher to design new architecture of the generation system.

The generation process for the research work in [4] consists of three main stages: morphological generation of lexical words, function words insertion, and syntax planning. In the first stage, the paper takes into consideration the morphological nature of Hindi, that Hindi nouns inflect for number and case; adjectives for case, number, and gender; verbs inflect based on GNPTAM (Gender-Number-Person-Tense-Aspect-Mood) information, voice, and vowel ending. Also, verb agreement with nouns are considered. In the second stage, UNL encodes case information by using relation labels assigned as per the properties of the connected nodes. Given a node along with all its lexical attributes from the UNL-Hindi dictionary, appropriate function words like-conjunctions, disjunctions, case marker, particles, etc., are inserted. In the third stage, syntax planning module linearizes the lexemes into the Semantic hyper-graph to make the word-order in the language flexible. Rule based approach to insert function words and heuristic rules for graph repair has been used. They used 901 Hindi sentences from agricultural domain to evaluate their system. A good correlation between fluency and the BLEU (bilingual evaluation understudy) scores, and strong correlation between fluency and adequacy scores is observed.

Keshari and Bista [40] proposed the architecture and design of UNL Nepali DeConverter for ‘DeCo’ tool. The proposed system has two major modules, namely, syntax planning module and morphology generation module. The syntax planning module is responsible for Nepali sentence formation by syntax planning. The morphology generation module is responsible for proper word formation through morphology generation. This module handles noun, verb and adjective morphology generation. Due to lack of standard UNL test data the system has yet not been formally evaluated. The size of the dictionary is small (only about 500 entries).

Martins *et al.* [41] proposed a prototype system for converting Brazilian Portuguese into the UNL and DeConverting UNL expressions into Brazilian Portuguese with ‘EnCo’ and ‘DeCo’ tools provided by UNDL foundation. Their system consists of three important sub-modules, namely, the lexical, the syntactic and the semantic modules. It requires inputs from a human expert who is seldom available and as such their performance is not quite adequate. They proposed the ‘*HERMETO*’ system which converts English and Brazilian Portuguese into UNL. This system has an interface with debugging and editing facilities along with its high level syntactic and semantic grammar that makes it more user friendly. There is no clear explanation how their system is evaluated and the results found are not discussed.

Dhanabalan and Geetha [42] proposed a DeConverter for Tamil language. It is a language independent generator that provides synchronously a framework for word selection, morphological generation, syntactic generation and natural collocation necessary to form a sentence. The proposed system involves the use of language specific, linguistic based DeConversion rules to convert the UNL structure into natural language sentences. This paper does not include how the system is evaluated and what results are gained.

## **3.2 Application of UNL**

Researchers have explored the use of UNL in different contexts such as machine translation in knowledge representation, knowledge management, multilingual search engine, language independent Universal Digital Library, etc. In this Section, we present the initiatives of researchers in this direction.

Alansary *et al.* [25] proposed the concept of language-independent Universal Digital Library within UNL framework. A UNL based Library Information System has been implemented as a

proof of concept. The UNL Library Information System provides access to the books of different languages into the native language of the user. The users can search the book catalog by entering a query in their natural language. The query is EnConverted to UNL and the search is performed. The search results contain the metadata of the books matching the query. These search results are then DeConverted by invoking the corresponding Language Server and displayed to user in his/her native language. Furthermore, the contents of the books are also converted to UNL and added to the UNL Encyclopedia, which contains the set of all UNL documents. The contents of these books can also be DeConverted and displayed in a natural language.

Mukerjee *et al.* [26] used UNL as a tool for language-independent semantics for Question Answering system. This concept has further been explored by Shukla *et al.* [32] suggesting two major tasks in the implementation of proposed Question Answering system. The first task is document processing that converts the document into equivalent UNL expression. For this, the document is processed through a UNL Parser to get the complete set of semantic predicates for the document. The other task is query processing and answer extraction. It involves the tasks of question analysis and processing to generate the answer template, conversion of natural language answer template to the natural language semantic predicates, conversion of natural language semantic predicates to Interlingua, and search engine and the conversion of the answer from the Interlingua to natural language through the DeConversion process.

Bértoli *et al.* [27] proposed ‘*CELTA*’ showcase as a web platform for using UNL. They demonstrated a multilingual business-to-business platform using UNL. Surve *et al.* [28] have proposed an ‘*Agro-Explorer*’ as a meaning based, Interlingua search engine designed specifically for the agricultural domain covering English, Hindi and Marathi languages. The system involves the use of ‘*EnCo*’ tool for the EnConversion of English query to UNL. The query in the UNL expression searches the UNL corpus of all documents. When a match is found, it sends the corresponding UNL file to DeConverter to provide the contents in the native language. Ramamritham *et al.* [29] further improved ‘*Agro-Explorer*’ to develop ‘*aAQUA*’ an online multilingual, multimedia agricultural portal for disseminating information from and to rural communities. The ‘*aAQUA*’ makes use of novel database systems and information retrieval techniques like intelligent caching, offline access with intermittent synchronization, semantic-based search etc.

Shaalán *et al.* [6] developed syntactic generation of Arabic. In the paper, they applied the concept of Arabic morphological and syntactic generation using grammar rules and Interlingua. That makes the research strong because it is easy to incorporate domain knowledge as well as heuristic rules into the linguistic knowledge. It also provides a high accuracy during generation.

Alansary [22] proposed language independent methodologies. The author considered the three acknowledged approaches in the area of machine translation. The author also examined other Interlingua based systems in detail by briefly describing their design and characteristics. The aim of the research was to bring down the language barriers. In addition, the results of their examination become a good contribution to newly modified UNL representation.

Uchida and Zhu [19] examined the necessity of Interlingua for multilingual translation system. The authors emphasized that Interlingua can localize the development of a machine translation system for a language since it has separate analysis and generation systems. The developers of analysis and generation system only needs to know the language and the Interlingua. The authors argued that Interlingua makes the common utilization of the knowledge necessary for machine translation systems, which helps to eliminate the redundancy to develop the same knowledge in different format.

Tanaka *et al.* [43] developed a multi-lingual machine translation system using Interlingua approach for five languages, namely, Chinese, Indonesian, Malaysian, Thai and Japanese. Their system had six important elements that include input system, sentence analysis system, Interlingua, electronic dictionary system, sentence generation system and output system. The research and development period is scheduled for six years, which is broken down roughly into three stages. In the first stage, a basic dictionary of 5,000 words are created and grammar test produced based on about ten sample sentences. In the second stage, the basic dictionary is expanded to 50,000 words, and technical term dictionary of 12,500 words are incorporated. In the third stage, the technical term dictionary is expanded to 30,000 words. Using the system developed in the previous stage, a large-scale translation experiment is done repeatedly to improve the dictionaries, the grammar, and the overall system. Finally, the system translates 5,000 words per hour with an accuracy of more than 90%, provided that the original texts are grammatically correct and all the words in the original texts are contained in the system dictionary.

Blanc [44] performed the integration of *'Ariane-G5'* to the proposed French EnConverter and French DeConverter. *'Ariane-G5'* is a generator of MT systems. In the proposed system, EnConversion takes place in two steps; first step is analysis of the French text to produce the representation of its meaning in the form of a dependency tree and the second step is lexical and structural transfer from the dependency tree to an equivalent UNL graph. Its DeConversion process also takes place in the two steps. The first step is lexical and structural transfer from the UNL graph to an equivalent dependency tree and the second step is the generation of the French sentence. This paper acknowledges that there are common work of the various UNL teams and the remaining discrepancies that may influence the quality of the processing. Also, it specifies an agreement about the correct use of the language but it does not include how the system is evaluated and what results are gained.

Daoud [45] proposed an Arabic DeConversion system which involves mapping of relations, lexical transfer, word ordering, and morphological generations. In the mapping of relations phase, each UNL relation has been mapped to the corresponding Arabic grammar structure which is implemented by DeConversion rules. Its word ordering phase is governed by DeConversion rules which are used during the insertion of a new node from the graph to the node-list. Arabic morphological generations are achieved by implementing a modular approach for coding the generation rules. During the development period of Arabic Module, the number of lexical items added to UNL-Arabic dictionary reached 120,000 entries. But, this paper does not include how the system is evaluated and what results are gained.

### **3.3 Summary**

In this Chapter, the research activities in the area of NLG have been documented. Based on the previous researches done, for sentence generation and multilingual machine translation system, Interlingua is always the first choice for researchers. The development of UNL as a standardized Interlingua has played a vital role in bridging the language barriers through multilingual NLG systems. In addition, UNL also emerged as an important standard for applications like knowledge representation, knowledge management, multilingual search engine, language-independent Universal Digital Library, etc. Even though the UNDL center came up with a tool called 'DeCo' for generating natural languages from UNL, problems have been identified regarding this tool.

Highly inflectional languages, like Arabic, have also been successful in developing sentence generator using UNL representation.

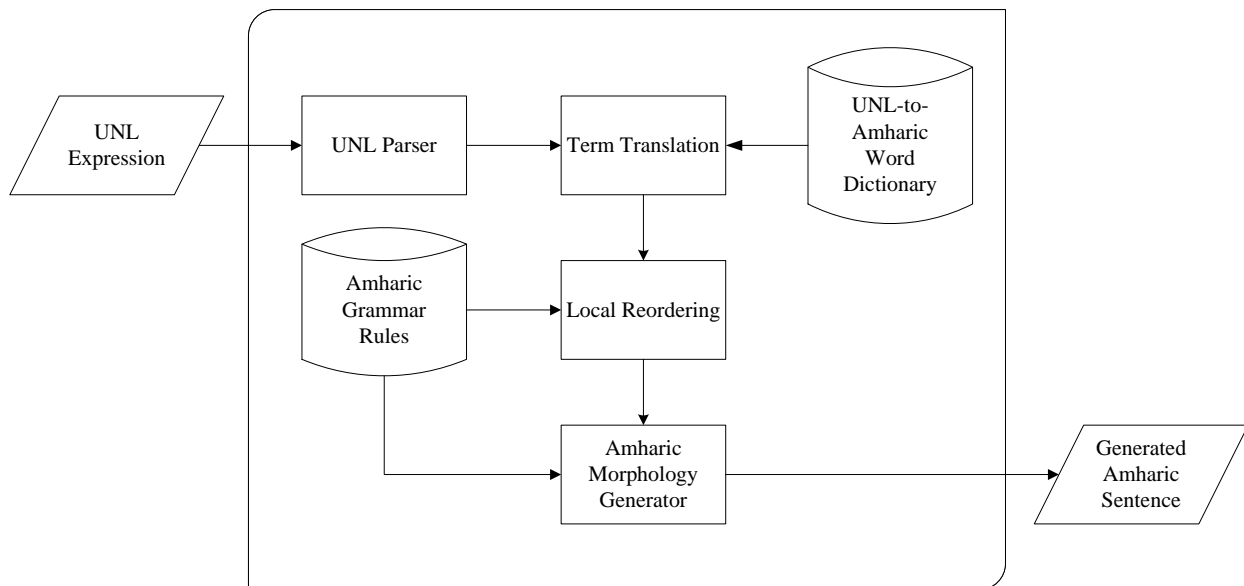
As a widely-spoken language of Ethiopia and as the country's working language, the Amharic language needs different works that are capable of interconnecting it to other languages. Even though various NLG systems have been proposed by different scholars for diverse languages, reusing what has been done for the other language is not adequate because of the dissimilar nature in natural languages. Moreover, the morphological complexity of a natural language could also force to implement a sentence generator for the corresponding language. Therefore, it is very important to develop a standardized NLG system for Ethiopian languages, particularly for Amharic. Such system would also be used for different application of NLP like CLIR, Cross-language Summarization, MT, DS and multilingual QA.

## Chapter Four: Amharic Sentence Generator

The main objective of this Chapter is to show the proposed Amharic sentence generator. In this Chapter the proposed architecture is described using different examples. Then, each module in the architecture is also discussed one by one. Finally, the summary of the Chapter is presented.

### 4.1 Architecture of Amharic Sentence Generator

The Amharic sentence generator generates an Amharic sentence from a given UNL expression. This process is based on the predicate-centric nature of UNL. A general architecture of Amharic sentence generator is given in Figure 4.1. The first stage of the generator is the UNL parser which parses the input UNL expression to build a node-net from the input UNL expression. The node-net created by the UNL parser is stored in a data structure which will be accessed by the remaining modules in the next processes. Afterwards, the Term Translation component searches each UW Amharic root word equivalent from the UNL-to-Amharic word dictionary, which is prepared for this purpose, and updates the data structure with Amharic root word and their grammatical attributes. Then, the local reordering module is used to define the word order in the sentence to be generated using the model prepared to arrange UNL expressions into Amharic. In addition, the marking order for UWs of a sentence are identified to make it ready for morphology generation. In Amharic Morphology Generation stage, the root words may be changed or something can be added or removed to get the complete sense of the words. Subsequently, the system makes use of morphology rules to generate a morphed word for those root words translated. Finally, function words or case markers, such as ላይ, ከ-, -ን, ለ, በ-, ወደ, *etc.* are inserted to the morphed words by the Morphological Generator. These function words are inserted in the generated sentence, based on the rule base designed for this purpose so that the output matches with a natural language sentence.



*Figure 4.1: Architecture of Amharic Sentence Generator*

The working of the generator is illustrated with the example sentence given in Listing 4.1 and this is used in the next subsections to describe how each module works.

```
{unl}
agt(kill(icl>do).@past.@entry, Peter(icl> person))
obj(kill(icl>do).@past.@entry, Mary(icl> person).@female)
ins(kill(icl>do).@past.@entry, knife(aoj> thing).@indef)
tim(kill(icl>do).@past.@entry, yesterday(aoj> thing))
rsn(kill(icl>do).@past.@entry, Jhon(icl> Person))
plc(kill(icl>do).@past.@entry, kitchen(icl>place).@def)
{/unl}
```

*Listing 4.1: UNL Expression Example Sentence*

The description and implementation details of each module of the proposed Amharic Sentence Generator is given in succeeding sections.

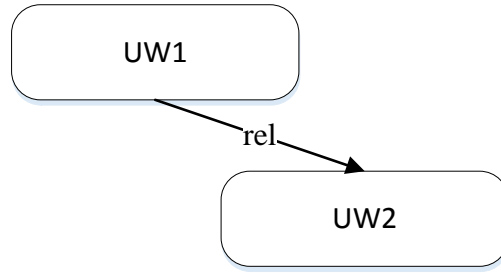
## 4.2 UNL parser

The purpose of a parser for natural language processing is generally to analyze sentences of a target language syntactically and semantically. Information on the syntax and grammatical features of each language is, therefore, necessary for a parser. Recently, attempts were made to develop parsers for annotated natural language texts in order to obtain more accurate results from sentences with ambiguities that cannot be solved by syntax and grammatical information only to mention few [46, 47].

However, the UNL Parser here needs to parse the UNL from input UNL expression without using language dependent grammatical information but only language independent annotations. Unlike current annotations, those for the UNL Parser are the set of tags required for generating the target meaning representations of sentences. The UNL Parser generates UNL expressions of the target meaning representations by interpreting such tags of annotation inserted in input sentences.

Thus, the UNL Parser can deal with any language because it does not require any language dependent grammatical information. It generates meaning representations of input sentences in any language by simply inserting tags for annotation in the input sentences.

The UNL parser is used to parse the input UNL expression to report the errors, if any, in the input expression. If input expression is in proper format or free from errors, then it builds semantic net known as node-net structure for input UNL expression, otherwise it reports the errors. This node-net is commonly called as UNL graph. The UNL graph consists of nodes and edges. A node in UNL graph represents a concept in the form of UW. An edge in node-net represents a UNL binary relation between two nodes. The edges in a UNL graph are directed from the parent node to child node. For a UNL binary relation  $rel(UW1, UW2)$ , UW1 is considered as parent of UW2. Thus, in a UNL graph the edge will be directed from UW1 to UW2 as shown in Figure 4.2. The UNL relation between UWs is used to label the edges between the nodes in the UNL graph, as shown in Figure 4.2.



*Figure 4.2: A UNL Graph Representation for UNL Binary Relation  $rel(UW1, UW2)$*

In this module, we adopt the algorithm of Singh *et al.* [4]. We developed a data structure that will subsequently be used in next phases of Amharic Sentence Generator. The procedure that has been used to create these data structures is given in Algorithm 4.1. The data structure created by this algorithm shall be accessed and modified by other modules of the Amharic sentence generator. The main aim of this linked list is creating nodes with a relation and attribute later on this nodes will easily be updated by other modules. Initially, this module only take care of the UWs, relations between the UWs and each UW attribute.

```

Input: UNL expression
Begin
  For (every relation in input UNL Expression)
    Node1 of UW ← UW1;
    Node2 of UW ← UW2;
    Relation ← Relation between UW1 and UW2;
    Node1 of Attribute ← UW1 attributes
    Node2 of Attribute ← UW2 attributes
    Node2 of Parent ← UW1
    If (Node1 not in NodeList)
      NodeList ← Node1;
    End-if
    If (Node2 not in NodeList)
      NodeList ← Node2;
    End-if
    RelationList ← Relation;
    End-if
  End-for
End
Output: Node List
         Relation List

```

*Algorithm 4.1: UNL Parser*

From Algorithm 4.1, Node1 and Node2 are two nodes of UNL graph, Relation represents relation between nodes, NodeList is a linked list of Nodes, RelationList is a linked list of Relations. Node is a data structure used to store an information about UW, HW, Attribute, Parent UW, etc. At this stage the parser only stores UW, Attribute, Parent UW and the relation that binds a UW with the Parent UW. Then, each module will update the initial information stored for every node in accordance with the changes made.

To illustrate with example, the UNL expression given in Listing 4.1 acts as input for the Amharic sentence generator. The UNL parser checks the input UNL expression for errors and generates the node-net or UNL graph as depicted in Figure 4.3 and node list as shown in Listing 4.2.

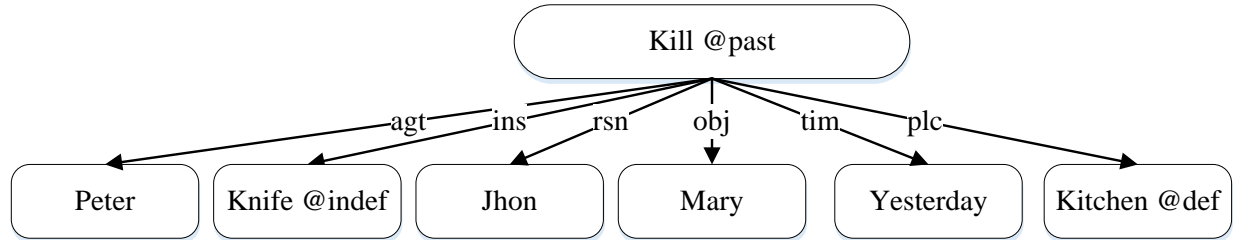


Figure 4.3: Simplified UNL Graph for UNL Expression given in Listing 4.1

```

Node1: Amharic word: UW: kill(icl>do).@past.@entry
Node2: Amharic word: UW: Peter(icl> person)
Node3: Amharic word: UW: Mary(icl> person).@female
Node4: Amharic word: UW: knife(aoj>thing).@indef
Node5: Amharic word: UW: yesterday(aoj>thing)
Node6: Amharic word: UW: Jhon(icl>Person)
Node7: Amharic word: UW: kitchen(icl>place).@def
  
```

Listing 4.2 : Node List Created by UNL Parser for UNL Expression Given In Listing 4.1

### 4.3 Term Translation

The Term Translation module translates UW from the parsed UNL sentence into an equivalent Amharic lexeme or HW. Here, every node in UNL graph is translated into an equivalent Amharic root word. This module handles the translation of words with their grammatical information attached. The result for the translation can easily be looked up from the UNL-to-Amharic word dictionary. But, this does not mean that this module completed its part. Rather, we need to filter out the translation results of each UW and update the attributes identified by UNL parser with Amharic syntactic properties.

During this phase, UWs are searched in the dictionary along with constraints specified in input UNL expression. A single word might have multiple sense and most UWs have attributes to restrict their senses. Using this restriction attributes as indicator, each node will be translated to Amharic lexeme or HW. Here, the restriction attributes of UW are used only to pick the correct sense. The other thing we need to consider is when we have more than one Amharic equivalent lexemes. In that case, this module takes frequency and flag value into consideration. For the possible HW, this module will rank and store the result in a temporary repository.

Besides, UNL-to-Amharic word dictionary incorporates information which are language dependent. Such information helps the generator to produce a sensible word for a sentence. Therefore, this module collects language dependent information from the UNL-to-Amharic word dictionary and appends it to the attributes of a node already created by the UNL parser. Such attributes are appended by preceding the ‘.@’ symbol.

The dictionary is suitable and in the required format for supporting the morphological, syntactic and semantic generation needed for language generation. The UNL-to-Amharic word dictionary uses a standard dictionary format from the UNDL foundation and stores three types of linguistic information. Morphological information is the information responsible for the correctness of the structure of words in Amharic language; these are word formation and different word forms, i.e., part of speech, lexical structure and inflectional paradigms. Syntactic information is the information responsible for generating well-formed Amharic structures, i.e., valence, aspect and sub categorization frames. Morpho-syntactic information is the information concerning grammatical categories and linguistic units that have both morphological and syntactic properties, i.e., transitivity, gender, number, etc.

An entry of the word dictionary contains three parts basically: a headword, a UW and a set of grammatical attributes (features). The UW of an entry expresses the meaning of its headword. Grammatical attributes define how a word or a morpheme behaves in a sentence.

Examples of entries of Amharic dictionary shown in Listing 4.3.

```
[ህዝብ] {} "people(icl>person)" (N, 3PL, ANIMT) <A, 0, 0>;
[አማርኛ] {} "Amharic (icl>language)" (N) <A, 0, 0>;
[እርሱ] {} "he(icl>male person)" (PRON, 3SG, MALE) <A, 0, 0>;
[ግድል] {} "kill(icl>do)" (V, VOA, A) <A, 0, 0>;
```

*Listing 4.3: Amharic Dictionary Sample Entries*

From Listing 4.2, the Term Translation module search the equivalent Amharic root words with grammatical attributes from the UNL-to-Amharic word dictionary and updates the node-list. Thus, the updated node-list is given in Listing 4.4.

```
Node1: Amharic word: ግድል; UW: kill(icl>do).@past.@entry.@V
Node2: Amharic word: ፒተር; UW: Peter(icl> person).@N.@PROP
Node3: Amharic word: ማሪ; UW: Mary(icl> person).@female.@N.@PROP
Node4: Amharic word: ቢላ; UW: knife(aoj>thing).@indef.@N
Node5: Amharic word: ጎሳጎት; UW: yesterday(aoj>thing).@N
Node6: Amharic word: ጆን; UW: Jhon(icl>Person).@N.@PROP
Node7: Amharic word: ኩሽና; UW: kitchen(icl>place).@def.@N
```

*Listing 4.4: Updated Node list form Listing 4.2*

## 4.4 Local Reordering

It is known that Amharic language is highly flexible. However, still some orders are considered more natural than others. Generally, the Amharic sentences are arranged in Subject, Object and Verb sequence. Hence, we assign relative positions to various words based on the relations they share with the HW in a phrase or sentence. Local Reordering deals with the arrangements of words in the generated output so that the output matches with the natural language sentence. It involves the matrix based priority of a relation; which would resolve the core challenges of positioning of HW, and the attributes needed for a HW generation by the Morphological Generator (the concept is similar with grammar model); which resolves a problem regarding gender, number, person, time, etc. conflict within a sentence.

In this module, two major techniques are applied for local reordering of HWs. The first one is using matrix based priority of a relation and the other one is using a simple grammar model for HW generation in which both techniques depend on Amharic language. A UNL binary relation plays a vital role in deciding which shall be ordered before which. As discussed in the UNL Parser section, from a UNL binary relation:  $rel(UW1, UW2)$ ;  $UW1$  acts as the parent of the relation whereas  $UW2$  acts as the child of the relation. For each parent child relationship, this module states whether the parent should be ordered before or after the child. We learn that in most of UNL relations, the children nodes appear to the left of the parent node in the generated output for Amharic.

To illustrate this concept, let us consider a UNL relation  $agt(UW1, UW2)$ , where  $UW1$  is a verb and  $UW2$  is the subject or agent of that event. Since Amharic is a SOV language, subject appears to the left of the verb in most cases. Same is the case of ‘obj’ relation.

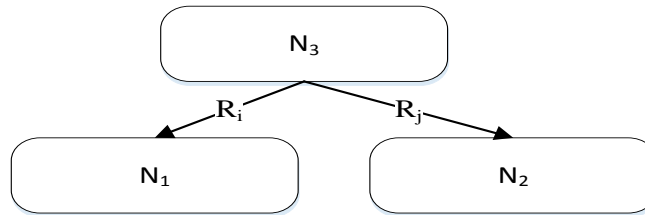
Therefore, from a UNL expression given in Listing 4.1 all the children, i.e., ‘Peter(icl>male person)’, ‘Mary(icl> person)’, ‘knife(aoj>thing)’, ‘yesterday(aoj>thing)’, ‘Jhon(icl>male person)’, and ‘kitchen(icl>place)’ will be placed to the left of the parent ‘kill(icl>do)’. The main question is how are we going to order the remaining UWs? Since all children are inserted to the left, the child inserted first in the generated output should be decided by some means. The proposed system uses the matrix based priority of relations to decide on this matter.

#### 4.4.1 Matrix Based Priority of Relations`

The necessity of matrix based priority of relations occurs when the child of two or more UNL relations has a common parent. It is important to decide the relative positions of children (sharing a common parent) with respect to each other in the generated output. In the proposed Amharic generator, the relative positions of children with respect to each other are decided by a matrix ‘M’. This matrix ‘M’ has 46 rows and 46 columns, representing 46 UNL relations specified in UNL specifications. This matrix  $M=[m_{ij}]$ ,  $i=1, 2, \dots, 46$ ;  $j=1, 2, \dots, 46$  contains the elements as ‘L’, ‘R’ and ‘-’. Here, ‘L’ means towards left, ‘R’ means towards right and ‘-’ means no action. If ‘ $m_{ij}$ ’ = ‘L’, then the position of the child of the  $i^{\text{th}}$  relation is left to the child of the  $j^{\text{th}}$  relation when the two children share a common parent. If ‘ $m_{ij}$ ’ = ‘R’, then the position of the child of  $i^{\text{th}}$  relation is right to the child of the  $j^{\text{th}}$  relation when the two children share a common parent. If ‘ $m_{ij}$ ’ = ‘-’,

then no action is to be taken as it is impossible that the child of this  $i^{\text{th}}$  relation shares a common parent with the child of this  $j^{\text{th}}$  relation.

This is illustrated with ‘ $R_i$ ’ and ‘ $R_j$ ’ as two UNL binary relations between three nodes ‘ $N_1$ ’, ‘ $N_2$ ’ and ‘ $N_3$ ’ as  $R_i(N_3, N_1)$  and  $R_j(N_3, N_2)$ . Here, node ‘ $N_1$ ’ and node ‘ $N_2$ ’ are the children of same parent ‘ $N_3$ ’ as shown in Figure 4.4.



*Figure 4.4: UNL Graph of Two Nodes with Same Parent*

Based on the structure of the target language, if ‘ $N_1$ ’ appears to the left of ‘ $N_2$ ’ in the generated sentence as denoted by ‘ $(N_1 L N_2)$ ’, then the priority matrix shown in Table 4.1 needs to be maintained for local reordering.

*Table 4.1: Matrix Representation for  $(N_1 L N_2)$  Structure*

	$R_i$	$R_j$
$R_i$	-	L
$R_j$	R	-

The priority of the child of the relation depends upon the frequency of ‘L’ in its row. If a relation has all ‘L’ in its row, then the child node of that relation will have highest priority and it will appear at the extreme left in the generated output. Similarly, if a relation has all ‘R’ in its row, then the child node of that relation will have lowest priority and will appear at the extreme right of all the children sharing the common parent in the generated output. Owing to the fact that we can associate a priority to the child associated with a relation, the matrix ‘M’ is called as Priority Matrix for Amharic and the detail of this is given in Annex B. This concept is illustrated with an example UNL expression given in Listing 4.1.

Considering the Amharic language structure, the priorities of each relation in UNL expression Listing 4.1 is as shown in Figure 4.5. Thus, the priority of ‘agt’ is greater than the priority of ‘obj’,

‘ins’, ‘tim’, ‘rsn’ and ‘plc’. This means the relation ‘agt’ is ordered to leftmost of a sentence to be constructed. The remaining relations are ordered following the values given by the priority matrix.

	agt	obj	rsn	tim	plc	ins
agt	-	L	L	L	L	L
obj	R	-	L	L	L	L
rsn	R	R	-	L	L	L
tim	R	R	R	-	L	L
plc	R	R	R	R	-	L
ins	R	R	R	R	R	-

High  
 ↓  
 Low

*Figure 4.5: Priority Matrix for Reordering of UNL Expression Given In Listing 4.1*

From Figure 4.5, ‘agt’ relation has the highest priority; counting the number of ‘L’ for each relation, i.e., 5; the ‘obj’ relation has priority 4; the ‘rsn’ relation has priority 3; the ‘tim’ relation has priority 2; while ‘plc’ and ‘ins’ have priority of 1 and 0 respectively. Thus, the Local Reordering module considers the attributes from the UNL sentence and reorder the child UWs starting with the highest priority relation from left to right. The parent node is processed to generate the correct word order after processing all the children nodes.

```
Peter(icl>male person) Mary(icl> person) Jhon(icl>Person)
yesterday(aoj>thing) kitchen(icl>place) knife(aoj>thing) kill(icl>do)
```

After local reordering is completed by applying priority of a relation, the next step of this module shall be identifying the attributes for a HW generation. For instance, if a HW is an Amharic verb and could be inflected for a subject’s number and object’s definiteness, here we will identify those properties before passing it to the Amharic Morphology Generation phase. To minimize the erroneous output sentence, this module performs marking order that answers the question of which HW shall be inflected by which attributes, because, in most cases attributes of a HW might conflict in number, gender, aspect, etc. with other HW in a sentence.

## 4.4.2 Amharic Marking Order

All Amharic verbs agree with their subject; and the subject of the verb is marked by suffixes or prefixes on the root verb. Object agreement is most often indicated by an object pronoun suffix because, unlike the markers of subject agreement, they do not vary significantly with the tense, aspect or mood of the verb.

Amharic verbs agree with their subjects or with their objects. They take the default form- singular, masculine when all nouns are oblique. In order to generate a verb form that is in concordance with the unmarked noun (subject or object), the noun's gender and number values are passed on to the verb's list of attributes. The rest of the information, i.e., for tense, aspect, mood, etc., is provided either by UNL attributes or by UNL-to-Amharic word dictionary.

Morphological rules generate morphemes for a verb that correspond to the value of these attributes. For example, a verb 'kill(icl>do)' with UNL attributes-@present and @progress, and with the attributes @female (from the subject or agent) and @pl (from the object) of the noun it agrees with, will be generated as- እየ- ገደል- ገደላችሁ (እየገደለችሁ). From the final output, we can understand the root verb 'ገደል' is inflected for the subject gender, which result in conjugating the morpheme '-ገደላችሁ' which shows subject agreement. From 'እየገደለችሁ' we know for sure the subject is feminine. Inflecting the verb for the object's number results in conjugating the morpheme '-ላችሁ'. In which the word becomes '(እየገደለችሁ)'. and we can tell the object is plural. This shows object agreement. Even though Amharic word generation is done by the morphological generator, this module identifies the attributes of nouns in which the verb agrees with.

To illustrate the concept further, consider the example shown in Listing 4.5:

```
agt (eat (icl>do).@entry.@past.@custom, he (icl>person).@respect)
obj (eat (icl>do).@entry.@past.@custom, bread.@def)
```

*Listing 4.5: UNL Expression Example to Illustrate Marking*

From Listing 4.5, unless we have identified the inflection attributes for a HW generation, we might get an erroneous sentence that conflicts with number. The updated nodes are shown in Listing 4.6 after the parsed UNL expression are searched by the Term Translation module.

```

Node1: Amharic word: ብል; UW: eat (icl>do).@entry.@past.@custom
.@V.@A
Node2: Amharic word: እርሳቸው; UW: he(icl>person).@respect.@PRON
Node2: Amharic word: ዳቦ; UW: bread.@def.@N

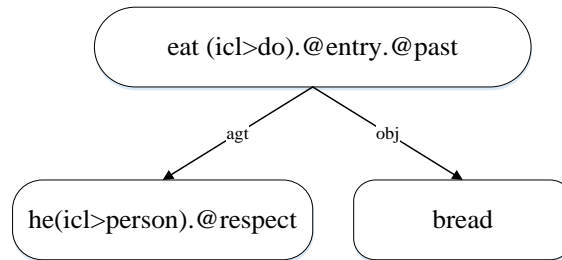
```

*Listing 4.6: Node List of Listing 4.6 after Term Translation is Processed*

The priority matrix and UNL graph for Listing 4.5 are shown in Figures 4.6 and 4.7 respectively.

High ↓ Low		agt	obj
	agt	-	L
	obj	R	-

*Figure 4.6: Priority matrix of Listing 4.5.*



*Figure 4.7: UNL Graph of Listing 4.5*

After applying a simple reordering with priority matrix as mentioned in Figure 4.6, the sentence would be ordered as follows:

he(icl>person) bread eat(icl>male person)

Note that the default values for generating a morphology are already set; for gender male, for number single, for voice active, for polarity positive, etc. Unless each HW has the attributes, the generation goes with the default values.

From Listing 4.5, the subject is ‘he(icl>person)’, the object is ‘bread’ and the verb is ‘eat (icl>male person)’. To illustrate how marking HWs are important for inflecting before Amharic word generation, let’s consider the following. During the generation process the HWs shall be inflected

based on their attributes. We call this attribute level resolution and this can be done using the rule base prepared for each part of speech for Amharic. Every HW could have both the UNL attributes and Amharic based attributes. As described in the previous section, the Amharic based attributes are stored in the dictionary and appended by Term Translation module for the corresponding node. Whenever attributes of a HW are not specified the generation continues with the default attributes. The system uses the attributes of a HW to generate an Amharic word. But, the final output of the generation could be grammatically unacceptable.

Therefore, by applying the generation of each node using only attributes of each UWs, the nodes is updated as shown in Listing 4.7:

```
Node1: Amharic word: በልቷል; UW: eat (icl>male
person).@entry.@past.@custom .@V.@A
Node2: Amharic word: እርሳቸው; UW: he(icl>person).@respect.@PRON
Node2: Amharic word: ዳቦውን; UW: bread.@def.@N
```

*Listing 4.7: Node List of Listing 4.6 after Morphology Generation is Processed*

Arranging the generated outputs as per the order gained is shown:

እርሳቸው ዳቦውን በልቷል

Separately observing at the phrases, “እርሳቸው በልቷል” there is a number conflict between the subject and the verb of the sentence; “ዳቦውን በልቷል” there is a conflict between the object and the verb of the sentence. Such conflicts occur because the generation of HWs is only based on attributes listed for each UW where no marking is applied.

However, instead of using attributes of a HW only, it identifies if HW can be inflected for number, gender, aspect, etc. of the object and the subject of the sentence. Depending on Amharic grammar rules, marking the verb for which attributes of the subject and the object to be inflected shall be resolved here. As depicted in Figure 4.8, the system will identify for which HW attributes the verb will be inflected.



Figure 4.8: Amharic Verb Inflection Structure

In order to overcome the subject-verb disagreement in the generated sentence this module used the following technique. Since Amharic verbs always agree with their subject or with the object, number attribute of a subject and number and definiteness attribute of object are marked for the verb. In Amharic language, if the subject of a sentence is referred in politeness/respect the number form of the subject is considered as plural. Always the number form of the pronoun or the subject would override the number form of the verb. Therefore, the attributes of the nodes are updated as in Listing 4.8.

```
Node1: Amharic word: ብል; UW: eat (icl>male person).@entry
.@past.@custom.@V.@A.@pl.@obj_def
Node2: Amharic word: እርሳቸው; UW: he(icl>person).@respect.@PRON
Node2: Amharic word: ዳቦ; UW: bread.@def.@N
```

Listing 4.8: Node List of Listing 4.6 after Marking of HWs is Processed

Attribute ‘.@pl’ and ‘.@def’ are the attributes marked for the verb ‘eat (icl>male person)’. The attribute ‘.@pl’ is marked for the verb because the subject ‘he(icl>person)’ is mentioned with respect. In Amharic, if the subject is referred with respect the verb should be inflected for plural number. As depicted in Listing 4.9, ‘.@obj\_def’ is added due to the objects ‘.@def’ attribute. The Amharic Morphology Generator updates the nodes as in Listing 4.9.

```
Node1: Amharic word: በልተውታል; UW: eat (icl>male person).@entry
.@past.@custom.@V.@A .@pl.@obj_def
Node2: Amharic word: እርሳቸው; UW: he(icl>person).@respect.@PRON
Node2: Amharic word: ዳቦውን; UW: bread.@def.@N
```

Listing 4.9: Node List of Listing 4.6 after Morphology Generation is Processed

Arranging the generated outputs as per the order identified previously results as follows:

እርሳቸው ዳቦውን በልተውታል

During the generation of ‘ብል’ the generator only cares about the following attribute- ‘.@entry.@past.@custom.@V.@A.@pl.@obj\_def’. The attribute ‘.@pl’ is gained from the subject’s ‘.@respect’. So, the generation becomes ‘ብልተ’. In the word ‘ብልተ’, the letter "ተ (-ትኸ)" explains either the subject is plural in number or referred with politeness. The attribute ‘.@def’ is another attribute from object of the sentence. The generation becomes ‘ብልተውት’. Due to inflecting the verb for ‘.@obj\_def’ attribute, the definite marker ‘-ው’ is added to the root word- ‘ብል-ትኸ-ው’. The number of the object is singular and masculine because there are no ‘.@obj\_pl’ and ‘.@obj\_female’ attributes. This results in fusing ‘-ት’ instead of ‘-ኣት’ or ‘-ኣቸው’ to the root word - ‘ብል-ትኸ-ው-ት’. Finally, by considering the verbs attribute ‘.@past.@custom’, the generator produces a sensible word ‘ብልተውታል’ by conjugating a morpheme ‘-ኣል’. Figure 4.9 is a summary for the above process.

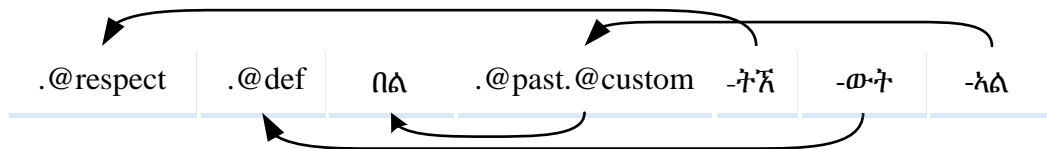


Figure 4.9: Inflection Marking Order

### 4.4.3 Reordering for simple sentences

A simple sentence contains a subject and a verb, and it expresses a complete thought. The UNL expression of simple sentences is converted into a simple node-net or UNL graph by UNL parser. The local reordering is handled in this module by processing the nodes sequentially. Algorithm 4.2 contains the instructions corresponding to this reordering.

```

Input: Node List
1. Begin
2. For (every node in the node list)
3.   If (node is unprocessed)
4.     TemRel  $\leftarrow$  node relation
5.     childNode  $\leftarrow$  UW
6.   Endif
7.   Set node to the next
8. End for
9. For (every pm in Priority Matrix)
10.  For (every relation in TempRel)
11.    If (pm matches with relation)
12.      OrderedNodelist  $\leftarrow$  corresponding childNode
13.    Endif
14.  Endfor
15. End for
16. Initialize Sentence
17. For (every activeNode in the OrderedNodeList )
18.  If (activeNode has no parent or contains '@entry')
19.    If (activeNode has no unprocessed child)
20.      Append the current HW to Sentence
21.      activeNode  $\leftarrow$  processed
22.      Break
23.    End if
24.    activeNode  $\leftarrow$  next unprocessed node of OrderedNodelist
25.  End if
26.  If (activeNode has parent)
27.    If (activeNode has no unprocessed child)
28.      Append the current HW to Sentence
29.      activeNode  $\leftarrow$  processed
30.      activeNode  $\leftarrow$  parent node of activeNode
31.    End if
32.    activeNode  $\leftarrow$  next unprocessed node of OrderedNodelist
33.  End if
34. End for
35. End
Output: Ordered Node List

```

*Algorithm 4.2: Reordering Algorithm for Simple Sentences*

Here, Algorithm 4.2 performs the processing of nodes by first identifying UNL relations and child nodes in a given expression. Then, it performs the processing of nodes according to priority matrix. Having the ordered node list according to the priority matrix, initializing of a sentence string follows. The given Algorithm loops over the highest priority child node sharing a common parent. Recursively it checks if a node has no further unprocessed children, then it is processed and added

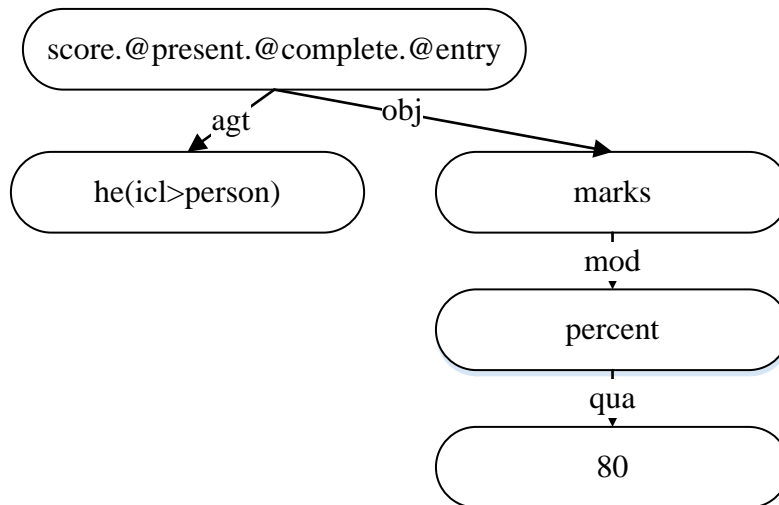
into the initialized sentence string. After processing of the child node, the parent of that node becomes active node and again traverses the highest priority unprocessed child node. This process continues until a node with no parent or the entry node gets processed.

The working of Algorithm 4.2 is illustrated with an example UNL expression given in Listing 4.10.

```
{unl}
agt(score.@present.@complete.@entry, he(icl>person))
obj(score.@present.@complete.@entry, marks)
mod(marks, percent)
qua(percent, 80)
{/unl}
```

*Listing 4.10: UNL Expression Example to Illustrate Simple Sentence Reordering*

UNL graph for UNL expression Listing 4.10 is depicted in Figure 4.10



*Figure 4.10: UNL Graph for UNL Expression in Listing 4.10*

The node list for the UNL expression given in Listing 4.10 after processing of Term Translation is given in Listing 4.11.

```

Node1: HW: ምዝግብ; UW: score.@present.@complete.@entry
Node2: HW: እርሱ; UW: he
Node3: HW: ውጤት; UW: marks
Node4: HW: ከመቶ; UW: percent
Node5: HW: 80; UW: 80

```

*Listing 4.11: Node List of Listing 4.10 after Term Translation is Processed*

As mentioned in matrix based priority of relations, the matrix is needed only for relations that have same parent. From Figure 4.10, the relations that have common parent is only 'agt' and 'obj'. The priority of the 'agt' is higher than the priority of the 'obj' relation as shown in Figure 4.11. Thus 'he' node will be traversed first. Note that the priority matrix given in Figure 4.11 is a sub-matrix of the earlier defined matrix 'M'.

High ↓ Low		agt	obj
	agt	-	L
	obj	R	-

*Figure 4.11: Priority Matrix for 'agt' and 'obj' Relations*

From the first for loop of Algorithm 4.2, temporary relation: 'agt', 'obj', 'mod' and 'qua'; and child nodes: 'he(icl>person)', 'marks', 'percent' and '80' are identified. The next for loop arranges temporary relations in accordance with priority matrix. During the relations sorting, it also sorts the order of the corresponding children nodes. Now, an ordered list of nodes is ready for further processing. From the ordered list, the first node is the 'he(icl>person)' node. It checks if 'he(icl>person)' node has no parent or if it contains '@entry' node. Since the 'he(icl>person)' node has parent and does not contain '@entry', the system checks if it has unprocessed children and the 'he(icl>person)' node appears to have no unprocessed child. Then, it will be processed and its HW is appended to the initialized sentence string used to store generated output, i.e., the sentence string will become 'እርሱ'. Then after, the active node will be changed to the Parent UW of the 'he(icl>person)' node. Now, parent of

'he(icl>person)' node, i.e., 'score' node will become the active node. It has only one unprocessed child, i.e., 'marks' node. So, it will be traversed next. It has one unprocessed child node 'percent', so it will be traversed. The 'percent' node also has one unprocessed child node '80', so it will be traversed next. The node '80' has no further unprocessed children, so it will finally be processed and its Amharic equivalent word will be appended to the sentence string, i.e., the sentence string will become 'እርሱ 80'. The parent of '80' node, i.e., 'percent' node now will become active node. The 'percent' node has no unprocessed children, so it will be processed and its Amharic equivalent word will be appended to the sentence string, i.e., the sentence string will become 'እርሱ 80 ከመቶ'. Now, the parent of 'percent' node, i.e., 'marks' node, will become an active node. The 'marks' node has no unprocessed children, so it will be processed and its Amharic equivalent word will be appended to the sentence string, i.e., the sentence string will become 'እርሱ 80 ከመቶ ውጤት'. Now, the parent of 'marks' node, i.e., 'score' node, will become an active node. It is a node with '.@entry' and has no unprocessed child, so it will be processed and its HW will be appended to the sentence string, i.e., the sentence string will become 'እርሱ 80 ከመቶ ውጤት ምዝገብ'. Since, the entry node is processed, the control will exit from the loop and the sentence output according to the local reordering will be available in the sentence string.

Thus, Algorithm 4.2 will result the processing of node-net in the order given:

Node2 Node5 Node4 Node3 Node1

The generated output of the proposed Amharic Sentence Generator after processed by the Amharic Morphological Generator will be:

እርሱ 80 ከመቶ ውጤት አስመዘገበ

From our first example in Listing 4.1, this module traverses the nodes given in Listing 4.4 based on the priority matrix given in Figure 4.5 in accordance with the Algorithm 4.2. Thus, the system yields the following result.

Node<sub>2</sub> Node<sub>3</sub> Node<sub>6</sub> Node<sub>5</sub> Node<sub>7</sub> Node<sub>4</sub> Node<sub>1</sub>

ፒተር ሜሪ ጆን ትላንት ኩቨና ቢላ ግድል

#### 4.4.4 Local Reordering for Compound Sentence

Scope is used to represent a compound universal word or a compound concept. A compound concept is a set of binary relations that are grouped together to express a complex concept. Local Reordering for sentences with compound concept is a bit different from simple sentences. In this case, UNL graph contains a scope node which itself is a UNL graph as depicted in Figure 4.12. Therefore, for UNL sentence with compound concept Algorithm 4.3 will be used.

```
Input: Node List  
Begin  
Initialize Sentence  
    If (UNL expression contains scope node)  
        Sentence  $\leftarrow$  Reorder consider scope node as a single  
        node with Algorithm 4.2;  
        TempSentence  $\leftarrow$  Reorder scope node using Algorithm 4.2;  
        Sentence  $\leftarrow$  Replace scope node by TempSentence;  
    End If  
End  
Output: Ordered Node List
```

*Algorithm 4.3: Reordering Algorithm for UNL graphs with scope node*

Local Reordering of UNL graph with a scope node is illustrated by an example UNL expression given in Listing 4.12.

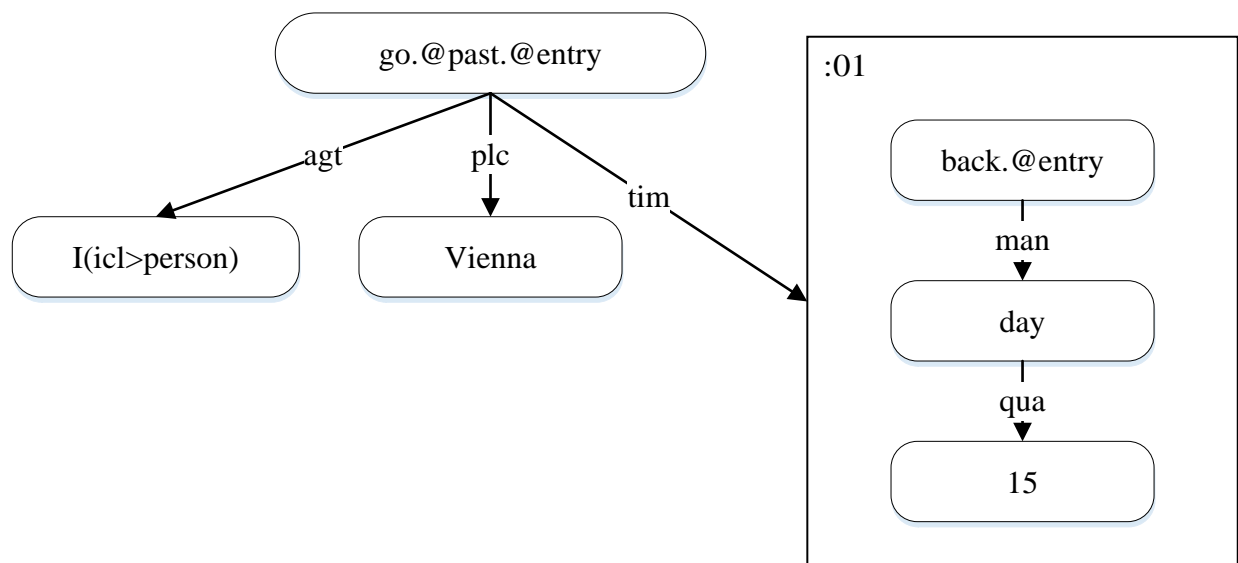
```

{unl}
agt(go.@past.@entry, I(icl>person))
plc(go.@past.@entry, Vienna)
tim(go.@past.@entry, :01)
man:01(back.@entry, day)
qua:01(day, 15)
{/unl}

```

*Listing 4.12: UNL Expression Example to Illustrate Compound Sentence Reordering*

UNL graph of UNL expression in Listing 4.12 is shown in Figure 4.12:



*Figure 4.12: UNL Graph with Scope Node for Listing 4.12*

From Figure 4.12, using Algorithm 4.3, 'go' is entry node having three children, 'I(icl>person)' with 'agt' relation; 'Vienna' with 'plc' relation and scope node ':01' with 'tim' relation. Considering the scope node ':01' as a single node it reorders the three children nodes using Algorithm 4.2. 'agt' relation has highest priority followed by 'tim' relation and then by 'plc' relation, as shown in Figure 4.13 of priority matrix. Thus, the system will first

traverse the 'I(icl> person)' node, followed by scope node ':01' and then by 'Vienna' node.

High		agt	tim	plc
↓	agt	-	L	L
	tim	R	-	L
Low	plc	R	R	-

Figure 4.13: Priority Matrix for 'agt', 'tim' and 'plc' Relations

The output of the first step of Algorithm 4.3:

I(icl>person) :01 Vienna go

The next step of Algorithm 4.3 is using Algorithm 4.2 to reorder the scope node. Then, the system will reorder the scope node ':01' and the output is:

15 day back

The final step of Algorithm 4.3 shall be replacing the scope node ':01' which is treated as a single node in the first output by the new output gained 15 day back. Finally, the result of Algorithm 4.3 is:

I(icl>person) 15 day back Vienna go

The Amharic sentence generated after the application of morphology generation is as:

እኔ ከ15 ቀን በፊት ከቪየና ሄድኩ

#### 4.4.5 Special cases in Local Reordering

In matrix based priority of relations, the priorities are assigned to relations of children nodes sharing a common parent. This approach does not consider the lexical properties of the children nodes. We have identified nine UNL relations which need to be treated as special cases during Local Reordering in Amharic language. These UNL relation are 'and', 'or', 'seq', 'fmt', 'equ', 'icl', 'nam', 'per', and 'cnt' relations.

For the above special cases, we propose to modify Algorithm 4.2 by adding Algorithm 4.4 between lines 24 - 25 and 31-32.

```
If (active node either of the special relation)
    Append the parent HW to Sentence
    activeNode ← special
End If
```

*Algorithm 4.4: Algorithm for Special Case Relations*

To illustrate the special case algorithm let's look at the following examples.

Example 1: UNL graph having relation 'and'

Let's consider UNL expression in Listing 4.13:

```
{unl}
agt(go.@future.@entry, I(icl>person))
plc(go.@future.@entry, church)
and(go.@future.@entry, pray)
ben(pray, you)
{unl}
```

*Listing 4.13: UNL Expression Having Relation 'and'*

UNL graph for UNL expression given in Listing 4.13 is depicted in Figure 4.10:

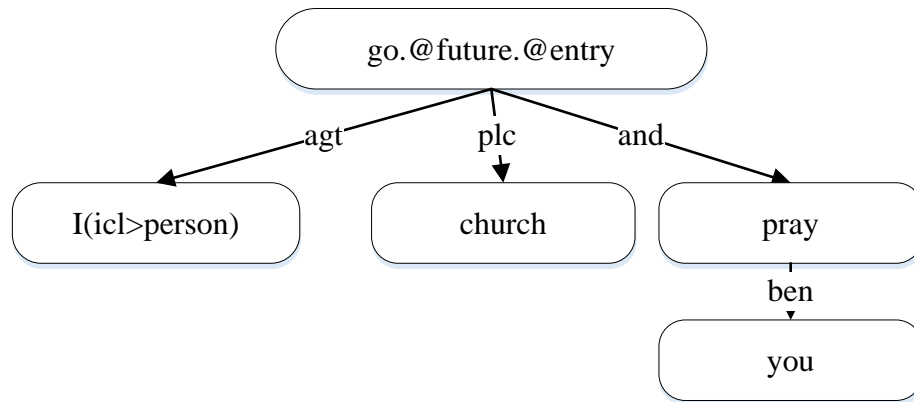


Figure 4.14: UNL Graph for UNL Expression Listing 4.13

The node list for the UNL expression given in Listing 4.13 after processing of Term Translation is given in Listing 4.14

```

Node1: HW: ህድ; UW: go
Node2: HW: እኔ; UW: I(icl>person)
Node3: HW: ቤተ ክርስቲያን; UW: church
Node4: HW: ጽልዬ; UW: pray
Node5: HW: አንተ; UW: you
  
```

Listing 4.14: Node List of Listing 4.13 after Term Translation is Processed

After apply Algorithm 4.2 and performing the Amharic Morphology Generation, the final output is:

እኔ ቤተ ክርስቲያን እና ለአንተ ጽልዬህ እሄዳለሁ

The output generated above is not acceptable order in Amharic. Considering the meaning, we can even say this order is wrong. The correct order could be as follows:

I(icl>person) church go you pray

To generate the correct Amharic syntax, we need to incorporate Algorithm 4.4 in our initial Algorithm 4.2. Therefore, if the system encounters relation ‘and’ as a next node of the ordered list, the parent node of relation ‘and’ shall be processed first and the child node is set as an active node.

From Listing 4.13, node ‘go’ has three children with ‘agt’, ‘plc’ and ‘and’ relations. According to the priority matrix, the ‘agt’ relation is of the highest priority, so the ‘I(icl>person)’ node shall be processed first since this node has no unprocessed children followed by node ‘church’ with a relation ‘plc’. Then, the system traverses to the node with a relation ‘and’. According to Algorithm 4.4 the system shall first process the parent node ‘go’; unlike the normal relations in which the child node is processed before the parent node. Its Amharic word will be appended into the sentence string and precede to the child node. Now, the child node of relation ‘and’ which is ‘pray’ becomes an active node. Since the ‘pray’ node has one unprocessed child, the active node becomes the ‘you’ node. Then the system processes the ‘you’ node and appends its Amharic equivalent into the sentence string and set the parent node to active node. Finally, the ‘pray’ node will be appended to the sentence string. Thus, the following Amharic sentence will be generated after application of Morphology Generation.

እኔ ቤተክርስቲያን እሄዳለሁ እና ለአንተ ጸልያለሁ።

Example 2: UNL graph having relation ‘or’

UNL expression is given in Listing 4.15

```
{unl}
obj(plant.@entry, tree.@pl)
scn(plant.@entry, direction.@pl)
mod(direction.@pl, different)
qua(tree.@pl, :01)
or:01(2.@entry, 3)
{/unl}
```

*Listing 4.15: UNL Expression Having Relation ‘or’*

The corresponding UNL graph is given in Figure 4.15.

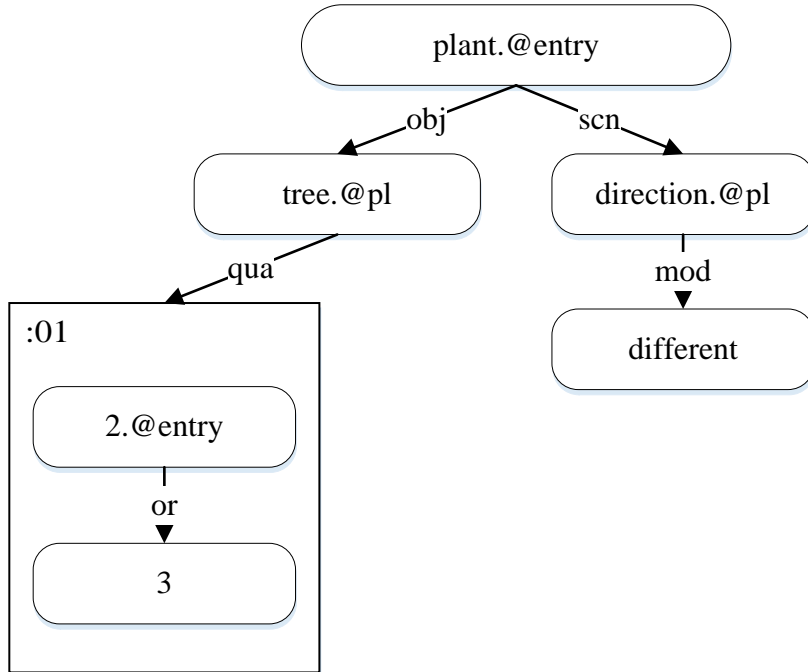


Figure 4.15: UNL Graph Having Relation ‘or’ for UNL Expression Listing 4.15

As mentioned above, using Algorithm 4.4 accordingly, in Figure 4.15 the parent of ‘or’ relation, *i.e.*, ‘2’ will be processed before its child node ‘3’. Since, the relation ‘obj’ has higher priority than relation ‘scn’ in the priority matrix, the reordering of UWs will be obtained as given below.

2 3 tree different direction plant

The output generated after application of morphology is:

2 ወይም 3 ዛሬች በተለያዩ አቅጣጫዎች ተከለ

The UNL graph consisting of the nine special relations has also been handled by the same strategy we used to resolve the ‘and’ and/or ‘or’ relation.

### 4.5 Amharic Morphology Generator

Morphology generation involves the mapping of the words proposed in the translation module to its more natural meaning. In this module, words are changed (something is added or removed) to get proper sense of words in terms of gender, number, tense, aspect and modality. The Amharic Morphology Generator module adapted the work of Markos Kassa Gobana [35], and customized morphology rules in accordance with UNL.

Here the Amharic Morphology Generator module accepts the HW, which is a root word in Amharic, with attributes stored in a temporary repository and inflect HW accordingly. During morphology generation, it uses the incorporated information within UNL attributes and grammatical information grabbed from the dictionary, which is also fused within the attributes. Thus, by combining with the grammatical rules, this module produces new lexemes that will fit the sentence to be generated.

This module uses Amharic grammar rules and critical information mentioned on UNL attributes and relations. Basically, we focus on noun and verb morphology rules. By taking the translated lexemes or HW from term translation module and referring their UNL attributes and a relation of a node, this module will inflect the lexemes accordingly.

The root words retrieved from UNL-to-Amharic word dictionary are inflected in this module depending on their gender, number, person, tense, aspect, and modality. This module also deals with the generations of articles in the target language. For example, definite articles (typically arise from demonstratives meaning ‘that’) are represented in UNL expression by the ‘@def’ attribute and it results in the generation of a new word. Similarly, indefinite articles (typically arise from adjectives meaning ‘one’) are represented by ‘@indef’ attribute and this results into the generation of the Amharic word አንድ(ዱ)/ አንዲት(ዋ).

```
{unl}
agt(eat(icl>do).@entry,child(icl>immature person).@def.@female)
obj(eat(icl>do).@entry,potato(icl>food).@pl)
plc(eat(icl>do).@entry,kitchen(icl>facilities).@def)
{unl}
```

*Listing 4.16: UNL Expression Example 2 Sentence*

From Listing 4.16, the Amharic Morphology Generator considers the attributes of the UNL expression to generate the morphed word. For instance, in the first UW ‘አጅ’ was the appropriate translation for UNL equivalent ‘child (icl>immature person)’ using our UNL-to-Amharic word dictionary. But if we still tried to take the UNL attributes into account, the correct

or convenient translation would be changed. From the UNL expression we find that ‘child (icl>immature person)’ is a female person from the UNL attribute ‘.@female’ that would make our translation be ‘ልጅት’ rather than ‘ልጅ’. In addition, we find an attribute that tells about whether the boy is known or unknown. It is described as ‘.@def’. Therefore, the morphological generator produces a word which will be more similar to the original context by applying morphological rules and the output of the generator will change to ‘ልጅቱ’. The same goes for UW knife, which is also definite, and the morphological generator will produce an output ‘ቢላው’.

The final output from this module are as follows:

```
child (icl> immature person).@def.@female to ልጅቱ
knife (icl> material).@def to ቢላው
```

Using attributes information this module is able to generate a word very close to its natural form. Having the general process described above in mind, here we will discuss how the morphology of Amharic words would be generated.

#### 4.5.1 Noun morphology

Noun morphology deals with the properties of nouns to identify their behavior in the generation process. In Amharic, nouns (N) inflect in number (singular or plural), species (definite or indefinite) and case (Nominative, Accusative, Genitive or Dative) while a given noun has an inherent gender, that is, no noun is both feminine and masculine at the same time.

In a UNL expression, information about plural number is represented with ‘@pl’ attribute. Absence of ‘@pl’ attribute conveys that the number being used is singular. For some lexemes like ‘ላም’, ‘ቦሬ’, ‘ንግሥት’, ‘ቁስ’, etc. the gender information are stored in the UNL-to-Amharic word dictionary. The morphological rules based on word paradigms generate a noun form using lexical, UNL relation and UNL attributes information. Morphology rules for Amharic language have been designed on the basis of their gender and the changes are made on its last character by using the orthographic operation. The format used for noun morphology rules is given below.

```
Insertion at Beginning, Orthography, Insertion at End, @Attribute1,
@Attribute2, ...
```

From the above format, 'Insertion at Beginning' represents character to be inserted at the beginning of the headword. 'Orthography' represents the orthographic operation to be applied and 'Insertion at End' represents character to be inserted at the end of headword and '@Attribute1, @Attribute2' and so on indicates the condition for the firing of the corresponding rule, i.e., morphology rule will be applied if attributes present in the rule base match with attributes of current UW in the UNL expression and its dictionary attributes. The noun morphology rule base is illustrated with an example rule base entry, given below.

null, orthography\_6\_3, ትዋን, @pl, @def, @female

Here, '@pl, @def, @female' is the list of attribute to be matched for the firing of this rule, i.e., if UW represents a noun which is definite with feminine gender and plural number, then corresponding HW changes its original form by inserting nothing at the beginning of HW, by applying an orthographic operation at the last character of the HW and by inserting 'ትዋን' at the end of HW. Let us consider a UNL expression involving UW with attributes as 'child(icl>immature person).def.@pl.@female'. The UW 'child(icl>immature person)' has 'ልጅ' as headword with '@pl', '@def' and '@female' attributes. Here, all the conditions for the firing of the above rule given are satisfied, and it will result into changing the last character from the sixth position to third and insertion of 'ትዋን' at the end of headword 'ልጅ' to form 'ልጅትዋን' as the morphed word for the given attributes.

The sample rule base for generation of noun morphology is given in Table 4.2. Depending on the gender of the noun the change is made on its last character by making use of the orthographic operations for the 16 cases.

*Table 4.2: Sample Noun Morphology Rule base*

No.	Noun Morphology Rules
1.	null, orthography_6_3, ትዋን, @pl, @def, @female
2.	Null, orthography_6_2, Null, @Sg, @Def, @Nom
3.	Null, orthography_6_2, ን, @Sg, @Def, @Acc
4.	የ, orthography_6_2, Null, @Sg, @Def, @Gen
5.	ለ, orthography_6_2, Null, @Sg, @Def, @Dat

6.	Null, Null, Null, @Sg, @Indef, @Nom
7.	Null, Null, ʔ, @Sg, @Indef, @Acc
8.	ʔ, Null, Null, @Sg, @Indef, @Gen
9.	ʌ, Null, Null, @Sg, @Indef, @Dat
10.	Null, orthography_6_7, ʔ orthography_6_2, @Pl, @Def, @Nom
11.	Null, orthography_6_7, ʔ orthography_6_2ʔ, @Pl, @Def, @Acc
12.	ʔ, Null, orthography_6_7, ʔ orthography_6_2, @Pl, @Def, @Gen
13.	ʌ, Null, orthography_6_7, ʔ orthography_6_2, @Pl, @Def, @Dat
14.	Null, orthography_6_7, ʔ, @Pl, @Indef, @Nom
15.	Null, orthography_6_7, ʔʔ, @Pl, @Indef, @Acc
16.	ʔ, Null, orthography_6_7, ʔ, @Pl, @Indef, @Gen
17.	ʌ, Null, orthography_6_7, ʔ, @Pl, @Indef, @Dat

An algorithm has also been developed to implement the noun morphology. This algorithm 4.5 is presented below.

```

Input: Node
Begin
    If (Number information is not present)
        Number ← singular
    End If
    Obtain HW
    Obtain Gender
    If (Gender information is not present)
        Gender ← male
    End If
    AttributeList ← HW, Number, Case, Species
    For (every rule in the noun morphology rules)
        If ( AttributeList match with Noun Morphology Rules)
            Fire rule
            Break
        End If
    End For
End
Output: Node

```

*Algorithm 4.5: Processing of Noun Morphology Rule base*

## 4.5.2 Verb Morphology

In Amharic, verbs are created mostly from consonantal radicals or roots which are inflected by a process of merging with vocalic components based on the various patterns which are sequences of vowels and consonants into which root consonants are inserted. In Amharic, even if majority of the verbs are tri-radical (three consonantal roots), the number of these radicals may range from two to five and more [34, 35].

Verbs in Amharic inflect in gender (male and female), person (1st, 2nd, and 3rd), number (singular or plural), Tense/Aspect/Mood (TAM) (perfect, imperfect, jussive /imperative, gerund, infinitive, participle, compound-perfect and the contingent that form verbs with auxiliaries) and voice (active and passive).

In this module, the verb morphology is generated on the basis of UNL attributes. The information about gender, person, number, etc. is generally extracted from subject of a sentence. For passive sentences, this information is extracted from the object of the sentence. In this situation, the UNL attribute '@passive' is associated with the main verb and '@topic' is associated with the object of the sentence.

*Table 4.3: Sample Parameter Value Passed to the Amharic Morphology Generator*

Rule No.	Amharic Verb	Verb morphology
1.	ሠብሮአል	(HW, PresPerf, positive, Type A, sg, male, act, Gerund, per3)
2.	ሠብራለች	(HW, PresPerf, positive, Type A, sg, female, act, Gerund, per3)
3.	ሠብረዋል	(HW, PresPerf, positive, Type A, pl, male, act, Gerund, per3)
4.	ሠብሮ ነበር	(HW, PastPerf, positive, Type A, sg, male, act, Gerund, per3)
5.	ሠብራ ነበር	(HW, PastPerf, positive, Type A, sg, female, act, Gerund, per3)
6.	ሠብረው ነበር	(HW, PastPerf, positive, Type A, pl, male, act, Gerund, per3)
7.	ይሠብራል	(HW, PresFut, positive, Type A, sg, male, act, Cont, per3)
8.	ትሰብራለች	(HW, PresFut, positive, Type A, sg, female, act, Cont, per3)
9.	ይሠብራሉ	(HW, PresFut, positive, Type A, sg, male, act, Cont, per3)
10.	እየሠበረ ነው	(HW, PresCon, positive, Type A, sg, male, act, Perf, per3)
11.	እየሠበረች ነው	(HW, PresFut, positive, Type A, sg, female, act, Perf, per3)
12.	እየሠበሩ ነው	(HW, PresFut, positive, Type A, sg, male, act, Perf, per3)
13.	ሥበር	(HW, null, positive, Type A, sg, male, act, Imperat, per3)

Rule No.	Amharic Verb	Verb morphology
14.	ሥበሪ	(HW, null, positive, Type A, sg , female, act, Imperat , per3)
15.	ሥበሩ	(HW, null, positive, Type A, pl , male, act, Imperat , per3)

Table 4.3 shows the sample parameters passed to the Amharic Morphology Generator. The decision is made based on Table 4.4. If a given verb contains matching attributes mentioned in one of the rows of Table 4.4, the rule shall be fired accordingly.

*Table 4.4: Sample Rule base for Verbs in Amharic.*

No	Tense	Aspect	Mood	Num	Gender	Per	Fire Rule	Verb	Suf/Pref
1.	.@present	.@custom	-	.@sg	.@male	.@3rd	#1	ሥበሮ	-አል
2.	.@present	.@custom	-	.@sg	.@femal e	.@3rd	#2	ሥበራ	-ለች
3.	.@present	.@custom	-	.@pl	.@male	.@3rd	#3	ሥበረ ው	-አል
4.	.@present	.@custom	-	.@pl	.@femal e	.@3rd	#3	ሥበረ ው	-አል
5.	.@past	.@complete		.@sg	.@male	.@3rd	#4	ሥበሮ	ነበር
6.	.@past	.@complete		.@sg	.@femal e	.@3rd	#5	ሥበራ	ነበር
7.	.@past	.@complete		.@pl	.@male	.@3rd	#6	ሥበረ ው	ነበር
8.	.@past	.@complete		.@pl	.@femal e	.@3rd	#6	ሥበረ ው	ነበር

No	Tense	Aspect	Mood	Num	Gender	Per	Fire Rule	Verb	Suf/Pref
9.	.@present	.@complete .@progress		.@sg	.@male	.@3rd	#7	ይሁብ ር	-አል
10.	.@present	.@complete .@progress		.@sg	.@femal e	.@3rd	#8	ትሰብር	-አለች
11.	.@present	.@complete .@progress		.@pl	.@male	.@3rd	#9	ይሁብ ር	-አሉ
12.	.@present	.@complete .@progress		.@pl	.@femal e	.@3rd	#9	ይሁብ ር	-አሉ
13.	.@past	.@complete	.@obl igatio n	.@sg	.@male	.@3rd	#13	ሥብር	
14.	.@past	.@complete	.@obl igatio n	.@sg	.@femal e	.@3rd	#14		-ኢ ሥብር
15.	.@past	.@complete	.@obl igatio n	.@pl	.@male	.@3rd	#15	ሥብር	-ኡ

### 4.5.3 Function Word Insertion

Function word insertion is used to insert function words like case markers and conjunctions in Amharic (e.g., ላይ, ከ-, ለ-, etc.) to the morphed words generated. Insertion of function words in the generated output depends upon UNL relation and the conditions imposed on parent and child nodes' attributes in a relation. The function words cannot be mapped directly to UNL relations due to following reasons. Firstly, a given UNL relation may have more than one function words corresponding to it. On the other hand, the position of inserting a function word in the generated output is not fixed. The function words are not simply inserted between the parent and child node

of a relation. These may be inserted before or after the parent/child node depending upon the conditions.

Therefore, we have identified rules to provide for the system on the varying nature function words insertions. This rule base consists of nine columns. The description of each column of this rule format is given below.

- First Column: Used to represent relation name
- Second Column: Used to represent function word preceding parent node
- Third Column: Used to represent function word following parent node
- Fourth Column: Used to represent function word preceding child node
- Fifth Column: Used to represent function word following child node
- Sixth Column: Used to represent positive conditions for parent node
- Seventh Column: Used to represent negative conditions for parent node
- Eighth Column: Used to represent positive conditions for child node
- Ninth Column: Used to represent negative conditions for child node

If there are more than one attributes that need to be asserted on a given node for firing of a rule, then they are stored in the rule base with the separation of ‘#’ sign. Here, the attributes represent the UNL attributes (obtained from given UNL expression) or the lexical attributes (obtained from UNL-to-Amharic word dictionary) of the node. The rule base for function word insertion is illustrated with an example rule given below

`obj:null:null:null:?:.@V:.@VINT:null:null`

At this point, ‘obj’ is a UNL relation under consideration, and the firing of given rule will result into the insertion of case marker ‘?’ following the child node in the generated output, because function word appears in the fifth column and second, third and fourth columns contain ‘null’ in the rule. The sixth column contains ‘.@V’, which means that the rule will be fired if the parent of ‘obj’ relation has a ‘V’ as its lexical attribute in the dictionary. The seventh column contains ‘VINT’ which refers to the attributes whose absence need to be asserted on the parent node for firing of the rule. It means that parent node should not contain ‘.@VINT’ (intransitive verb). The eighth column of the rule given contains ‘null’ which refers to the attribute, whose presence needs to be asserted on the child node for firing of the rule, *i.e.*, child node is free from restriction. Ninth

column contains 'null' which refers to the attribute whose absence needs to be asserted on the child node for firing of the rule. It means that child node is still free from restriction. Thus, if relation 'obj' has a parent node with '@V' attribute; and without '@VINT' attribute, then case marker ? will be inserted following the child node in the generated output.

Once again from Listing 4.1, here the morphological rules are applied to modify Amharic words stored in the nodes according to UNL attributes given in input UNL expression and dictionary attributes retrieved from UNL-to-Amharic word dictionary. The nodes given in Listing 4.4 are processed by Amharic morphology generator module. The processed nodes are given in Listing 4.17.

```

Node1: Amharic word: ገደላት; UW: kill(icl>do).@past.@entry.@V
Node2: Amharic word: ፒተር; UW: Peter(icl>male person).@N.@PROP
Node3: Amharic word: ማሪ; UW: Mary(icl> person).@female.@N.@PROP
Node4: Amharic word: ቢላ; UW: knife(aoj>thing).@indef.@N
Node5: Amharic word: ትናንት; UW: yesterday(aoj>thing).@N
Node6: Amharic word: ጆን; UW: Jhon(icl>Person) .@N.@PROP
Node7: Amharic word: ኩሽናው; UW: kitchen(icl>place).@def.@N

```

*Listing 4.17: Node List of Listing 4.4 after Morphology is Processed*

It is evident from nodes given in Listing 4.17 that, the Amharic Morphology Generator ግድል is changed to ገደላት and ኩሽና is changed to ኩሽናው by the Amharic Morphology Generator. This module is also responsible for the insertion of function words in the morphed lexicon as depicted in Listing 4.18.

```

Node1: Amharic word: ገደላት; UW: kill(icl>do).@past.@entry.@V
Node2: Amharic word: ፒተር; UW: Peter(icl>male person) .@N.@PROP
Node3: Amharic word: ማሪን; UW: Mary(icl> person).@female.@N.@PROP
Node4: Amharic word: በቢላ; UW: knife(aoj>thing).@indef.@N
Node5: Amharic word: ትናንት; UW: yesterday(aoj>thing).@N
Node6: Amharic word: በጆን ምክንያት; UW: Jhon(icl>Person).@N.@PROP
Node7: Amharic word: ኩሽናው ውስጥ; UW: kitchen(icl>place).@def.@N

```

*Listing 4.18: Node List of Listing 4.17 after Function Word Insertion is Processed*

The rules fired during the insertion of function words in Listing 4.18 are shown in Listing 4.19:

```

ins: null: null: በ: null: .@v: null: null: null
rsn: null: null: በ: ምክንያት: .@v: null: null: null
plc: ውስጥ: null: null: null: .@v#.@N: null: null: null

```

*Listing 4.19: Function Word Insertion Sample Rules7*

In this phase, case markers ን, በ, በ... ምክንያት and ውስጥ are added to Node<sub>3</sub>, Node<sub>4</sub>, Node<sub>6</sub> and Node<sub>7</sub>, respectively, according to the function word insertion rule base. Therefore, the final output is shown as follows:

ፒተር ማሪን በጆን ምክንያት ትላንት ኩሽናው ውስጥ በቢላ ገደላት

From the generated Amharic sentence given above, the system demonstrates that it is capable to convert the input UNL expression to Amharic successfully.

## 4.6 Summary

In this Chapter, we described how the UNL based Amharic sentence generator works. We have tried to explain each component in detail. The proposed architecture for Amharic sentence generator contains four components: UNL parser, Term Translation, Local Reordering and Amharic Morphological Generator. UNL-to-Amharic word dictionary is also prepared as a part of

our research work. We have tried to identify Amharic grammar rule for generating words as well as ordering the words according to the acceptable word order in Amharic. We have considered the UNL relation to prepare the priority matrix which is used as Amharic language model. The grammar model for Amharic is attempted to resolve by marking Amharic verbs for the subject and object of a given UNL sentence. The HW or Amharic word is generated by the Amharic Morphology Generator. Since the UNL parser module creates a node list for each UW and stores the attributes, the Amharic Morphology Generator uses the attributes of a HW to generate. For Amharic part of speech, we have prepared a rule base, basically for Amharic Nouns and Verbs. Thus, the generator matches the parameters and fire the rules accordingly. After the system gets the generated HWs with Amharic acceptable order, it uses the Amharic Morphology Generator module for further furnishing the sentence and increases the acceptability of the sentence.

# Chapter Five: Experiment

This Chapter presents the experiments on the work carried out in this thesis. The tools as well as the prototype development of the Amharic sentence generator have been presented. The evaluation has been discussed by first identifying test cases selection then presenting the evaluation methods for Amharic sentence generator and by showing the test results according to the tests selected. Finally, the results of this experiment are discussed.

## 5.1 Prototype Development

A prototype is developed to show the validity of the proposed Amharic sentence generator. We have developed a prototype which is capable to perform the generation with user interference. Thus, the system will allow users to upload a text file which contains the UNL expression and the system will process and show the generated output sentence.

The proposed system allows a user to either upload a text file or write the standard UNL expression as shown in Figure 5.1. Once the UNL sentence is uploaded, the UNL parser module starts to parse the contents of the expression and create a linked list for the nodes of the UNL graph.

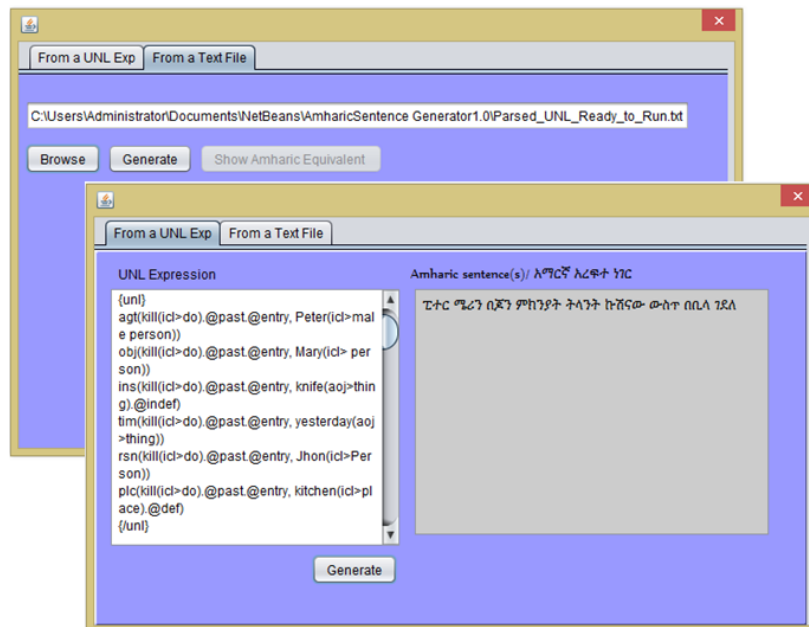


Figure 5.1: Amharic Sentence Generation GUI

Further processing will continue until the system generates the final output Amharic sentence. In case of the input UNL expression passed to the system through text file the output of our system shall be through a text file. Whenever the input is given directly to the system, the generated output shall be visible both in the system interface and text file.

Java programming language with NetBeans development environment is used in this research work. All the algorithms used in different modules of Amharic sentence generator have been implemented with Java to develop the proposed system. We have also used MySQL database for storing our UNL-to-Amharic word dictionary.

## **5.2 Evaluation**

### **5.2.1 Selection of Sentences for Test Cases**

An important aspect in NLG evaluation is to make appropriate selection of sentences for evaluating the generator. According to Balkan [48], there are three types of sources for evaluating a sentence generator. The first one is test corpora and it is a collection of naturally occurring text. The second one is test suites which is a collection of usually artificially constructed inputs, where each input is designed to examine a system's treatment to a specific phenomenon. The third one is test collections which is a set of inputs associated with a corresponding set of expected outputs. As a result, 142 UNL sentences have been selected for evaluation of the proposed Amharic sentence generator which are found from the UNDL foundation, ethics and science domain.

### **5.2.2 Evaluation of Amharic Sentence Generator**

The proposed Amharic sentence generator is evaluated by inputting UNL expressions selected for test cases. The selected UNL sentences have an incorporated original English version and the UNL sentences, which are translated by native Amharic speakers who have a very good skill of English and a good understanding of UNL. Thus, the output sentences of the Amharic sentence generator are compared with the sentences translated by those native speakers using the tests picked for evaluation. The two Amharic sentences, the sentence translated by the native Amharic speaker and the sentence generated by the generator, are compared and the system is evaluated based on different tests on these outputs.

### 5.2.3 Tests for the evaluation

There are a number of tests available for evaluating NLG systems. Slype [49] describes subjective tests and quantitative tests for evaluation. Subjective tests like adequacy and fluency tests have been performed on the proposed system. The quantitative test, which is error analysis, has also been performed by calculating Word Error Rate (WER) of the proposed system for Amharic language.

#### Fluency Test

Fluency refers to the degree to which the target is well formed according to the rules of target language grammar. A fluent segment is one that is well-formed grammatically, contains correct spellings, adheres to common use of terms, titles and names, is intuitively acceptable and can be sensibly interpreted by a native speaker of that language [50]. There are a number of discussions available in the literature regarding the choice for point of scale in the evaluation. It has been suggested that one can use 5-point scale for these tests [50]. Singh *et al.* [4], however, suggested that 5-point scale is a too fine-grained scale and in this scale the distinction may result in evaluators worrying a lot about making an accurate call, and intuitive judgment may get affected. It also makes the evaluation even more subjective. Thus, a 4- point scale has been followed for evaluation of the proposed system as followed by Singh *et al.* [4]. the details of 4-point scale for the fluency score is given in Table 5.1.

*Table 5.1: Four Point Scale of Fluency Score*

Score	Level	Description
4	Perfect	Good grammar
3	Fair	Easy-to-understand but flawed grammar
2	Acceptable	Broken, understandable with effort
1	Nonsense	Incomplete

#### Adequacy Test

Adequacy refers to the degree to which the information present in the original sentence is also communicated in generated sentence. Thus, for adequacy judgments, the gold standard will serve

as a proxy for the original source language text. For the evaluation of the proposed Amharic sentence generator, the sentence generated by the system is compared with the sentence translated by the native speaker, whose UNL expression is provided for the Amharic sentence generator. Thus, for the adequacy test the evaluator is also presented with Amharic native speaker translation, *i.e.*, the translation from the input UNL sentence, with reference to the system generated sentences which they can compare the adequacy of the output. Again, a 4-point scale has been followed for the adequacy test of the proposed system as followed by Singh *et al.* [4]. The details of 4-point scale for the adequacy score is given in Table 5.2.

*Table 5.2: Four Point Scale of Adequacy Score*

Score	Level	Description
4	All	No loss of meaning
3	Most	Most of the meaning is conveyed
2	Some	Some of the meaning is conveyed
1	None	Hardly any meaning is conveyed

### **Word Error Rate**

Word Error Rate (WER) performs error analysis to establish how seriously errors affect the generated output. WER is a quantitative metrics which is based on the analysis of errors made by the sentence generator. WER is defined as percentage of words which are to be inserted, deleted, or replaced during the generation in order to obtain the acceptable sense of the UNL expression [49].

### **5.3 Discussion**

The other important thing to consider is choosing appropriate evaluators for performing the tests on the proposed system. The proposed system has been evaluated by three evaluators having a good knowledge of Amharic. A good understanding of UNL expression was expected from the evaluators. We have tried to brief them the basic concept so that they will have the knowhow. As suggested by Ma X. and Cieri C. [50] and by Singh *et al.* [4], the evaluators were asked to provide their intuitive reaction to the output and to work as quickly as comfortable. First of all, fluency

judgments have been taken. The evaluators did not have any clue about the original source sentence for these judgments. After fluency judgments, the evaluators were asked to look at the UNL expression and original English sentences for adequacy judgments.

### 5.3.1 Results for fluency score

Among the selected 142 UNL expressions selected for evaluating the system, Table 5.3 shows the input UNL expression and the corresponding output of the system. Using the generated sentence the evaluators check the correctness of the grammar, the spelling, use of terms, titles and names, intuitively acceptable and sensibly interpreted.

*Table 5.3: Generated Amharic Sentence from UNL Expression Sample*

Input UNL Expression	Generated Amharic Sentence
{unl}and(science(icl>knowledge):0B.@entry, ethics(icl>principle):00) {/unl}	ግብረገብ እና ሳይንስ
{unl}mod(Director-General(icl>person):00.@entry, UNESCO(equ>United Nations Educational Scientific and Cultural Organization):0K) {/unl}	የተባበሩት መንግስታት ትምህርት ሳይንስ እና ባህል ድርጅት የበላይ ሥራ ኃላፊ
{unl} and(UNESCO(equ>United Nations Educational Scientific and Cultural Organization):28.@entry, education(icl>activity):1X) and(education(icl>activity):1X, technology(icl>knowledge):1L) and(technology(icl>knowledge):1L, science(icl>knowledge):1C) and(science(icl>knowledge):1C, progress(icl>phenomenon):12) and(progress(icl>phenomenon):12, development(icl>process):0P)	ግብረገብ ፣ ሥነ-ምግባር ፣ እድገት ፣ ለውጥ ፣ ሳይንስ ፣ ሥልጣኔ ፣ ትምህርት ፣ ተባበሩት መንግስታት ትምህርት ሳይንስ እና ባህል ድርጅት

Input UNL Expression	Generated Amharic Sentence
<p>and(development(icl&gt;process):0P, moral(icl&gt;virtue):0I) and(moral(icl&gt;virtue):0I, ethics(icl&gt;principle):0A) {/unl}</p>	
<p>{unl}man(put(icl&gt;express(agt&gt;thing,obj&gt;thing)): 08.@past.@entry, bluntly:00) obj(put(icl&gt;express(agt&gt;thing,obj&gt;thing)):08.@p ast.@entry, :01) aoj:01(absurd(aoj&gt;thing):0T.@entry, question(icl&gt;matter):0H.@topic.@def) {/unl}</p>	<p>ፍርጥ አድርገህ አስቀምጥ ጥያቄው የማይመሥል ነው</p>
<p>{unl} obj(represent(aoj&gt;thing,obj&gt;thing):0V.@entry.@ should.@past.@interrogative, force(icl&gt;armed force):05.@topic) aoj(represent(aoj&gt;thing,obj&gt;thing):0V.@entry.@ should.@past.@interrogative, time(icl&gt;abstract thing):0P.@pl) tim(represent(aoj&gt;thing,obj&gt;thing):0V.@entry. @should.@past.@interrogative, present time:18.@pl) mod(time(icl&gt;abstract thing):0P.@pl, future(mod&lt;thing):0I) mod(force(icl&gt;armed force):05.@topic, what(mod&lt;thing):00.@qfocus) {/unl}</p>	<p>መጪው ምንድን ነው በዚህ ጊዜ ጊዜ ኃይልን የወከለ ?</p>
<p>{unl} and(lie(icl&gt;remain(gol&gt;thing,obj&gt;thing)):5H.@e ntry, "":00) obj(lie(icl&gt;remain(gol&gt;thing,obj&gt;thing)):5H.@e ntry, this(icl&gt;thing):5C.@topic)</p>	<p>እና ይህ በእራሱ በሥነ-ምግባር ውስጥ ይገኛል</p>

Input UNL Expression	Generated Amharic Sentence
<p>gol(lie(icl&gt;remain(gol&gt;thing,obj&gt;thing)):5H.@entry, ethics(icl&gt;principle):5T)</p> <p>mod(ethics(icl&gt;principle):5T, it(icl&gt;thing):5P)</p> <p>{/unl}</p>	
<p>{unl}</p> <p>aoj(environment friendly(aoj&gt;thing):0N.@not.@entry, they(icl&gt;person):06.@topic)</p> <p>man(environment-friendly(aoj&gt;thing):0N.@not.@entry, all(icl&gt;how):0J)</p> <p>plc(environment-friendly(aoj&gt;thing):0N.@not.@entry, respect(icl&gt;aspect):1H.@pl)</p> <p>man(environment-friendly(aoj&gt;thing):0N.@not.@entry, but(icl&gt;how):00)</p> <p>mod(respect(icl&gt;aspect):1H.@pl, other(icl&gt;additional(mod&lt;thing)):1B)</p> <p>{/unl}</p>	<p>ሌላው መልኩ ሁሉም እነዚህ ለአካባቢ ተስማሚ አይደሉም</p>

The fluency score of the proposed system was 3.0 (on a 4-point scale). The response by evaluators is further analyzed and following are some important findings for the fluency test.

- 32.16% sentences got a score 4, *i.e.*, these are perfect having no grammatical mistake.
- 39.20% sentences got a score 3, *i.e.*, these are fair and easy to understand.

- 24.12% sentences got a score 2, *i.e.*, these are acceptable and are understandable with effort.
- 4.52% sentences got a score 1, *i.e.*, these are hard to understand.

Therefore, the proposed system generated 71.4% grammatically correct sentences. These sentences are those that have a score of 3 or above.

### 5.3.2 Results for Adequacy Score

The adequacy score of the proposed system was 2.9 (on a 4-point scale). The response by the evaluators has again been further analyzed and following are some important findings for the adequacy test.

- 32.66% sentences got a score 4, *i.e.*, there is no loss of meaning during their translation.
- 35.18% sentences got a score 3, *i.e.*, most of the meaning of these source sentences is conveyed in the translated sentence.
- 21.11% sentences got a score 2, *i.e.*, some of the meaning of these source sentences is conveyed in the translated sentence.
- 11.06% sentences got a score 1, *i.e.*, hardly any meaning of these source sentences is conveyed in the translated sentence.

Therefore, the proposed system generated 67.8% sentences that are faithful to the original UNL expression. These sentences are those that have a score of 3 or above.

To see if the fluency score have a correlation with adequacy score, we applied Pearson's Correlation Coefficient formula as shown in (5.1).

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (5.1)$$

Where:

- $\Sigma$  is Sigma, the symbol for "sum up"

- $(x_i - \bar{x})$  is each x-value minus the mean of x
- $(y_i - \bar{y})$  is each y-value minus the mean of y

We can obtain a formula for  $r$  by substituting estimates of the covariances and variances based on the results into the formula above. So we have one dataset for fluency score  $\{x_1, \dots, x_n\}$  containing  $n$  values and another dataset for adequacy score  $\{y_1, \dots, y_n\}$  containing  $n$  values. In our case since the number of UNL expression for evaluation is 142 the value of  $n$  is equal to 142. Finally, the result of  $r$  turn out to be 0.55.

Since the value of  $r$  is greater than 0, we can conclude that there is a positive correlation between fluency and adequacy scores. This relation between adequacy and fluency is explored further in Figure 5.2. It shows the distribution of adequacy scores for various values of fluency.

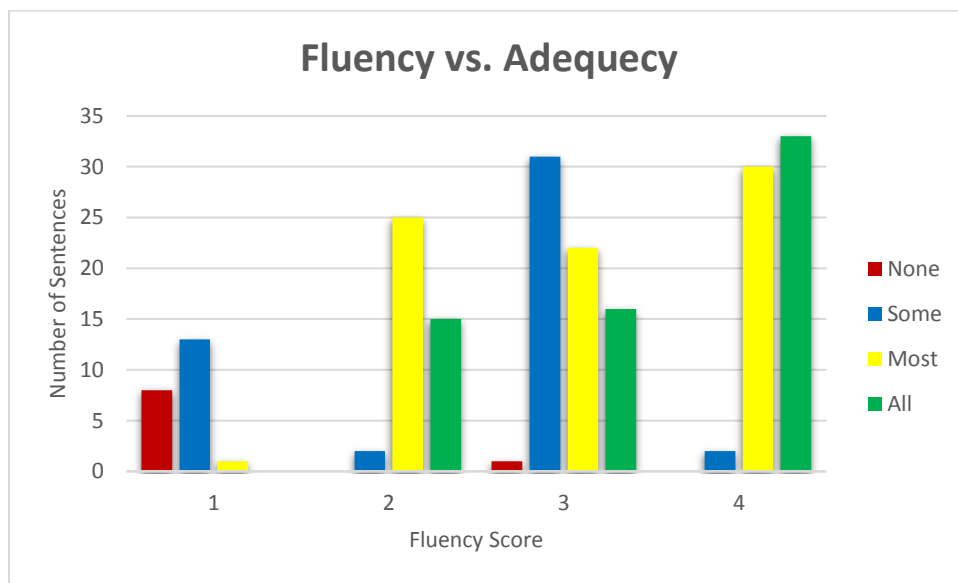
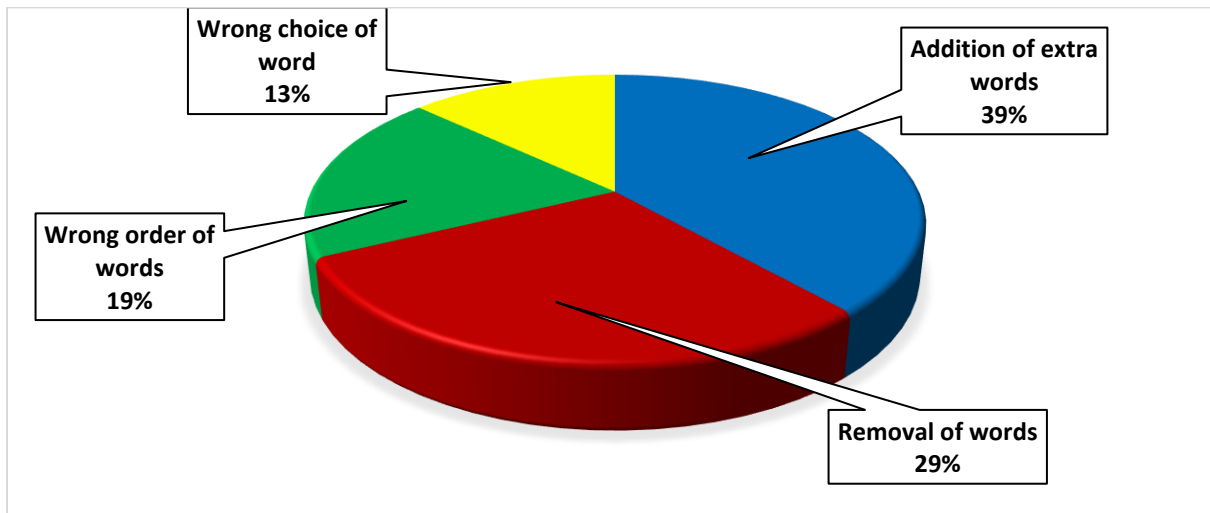


Figure 5.2: Distribution of adequacy scores for various fluency score.

### 5.3.3 Results to the Error Analysis

Error analysis has also been performed for the proposed system after calculating fluency and adequacy scores. The error analysis has been performed for classified error list that includes the errors like addition of extra words, removal of words, wrong order of words, and wrong choice of words in the generated output. All these errors in the generated output have been identified and

noted. After robust analysis, word error rate is found to be 28.94%. The results of word error rate are depicted in Figure 5.3.



*Figure 5.3: Word Error Rate Analysis of the Proposed System*

From Figure 5.3, it can be concluded that the majority of the errors are due to addition of extra words. The addition of extra words in the generated sentence is decided by the function word insertion part of Amharic Morphological Generator module. Since the function words are those words used to furnish the generated Amharic sentence, the errors due to this module will not severely affect the accuracy of the system. The order of words in the generated sentence is decided by the Local Reordering module of Amharic Sentence Generator. This order is governed by a matrix based on priority of UNL relations sharing a common parent. The Amharic morphology generation rules are responsible for removal of words in the generated output and these are under Amharic Morphology Generator module. The errors in this module arise from firing of inaccurate morphology generation rule. This mostly requires semantic information of verbs. The non-availability of this information for some verbs in UNL-to-Amharic word dictionary results into incorrect generation. In addition, the function word insertion also requires a well-formed rule base that could handle the different aspects of adding preposition and postposition. Since the rule base is not exhaustive enough, the results are not satisfying. Finally, Term translation module is responsible for the wrong choice of words in the generated output which have the least percentage of errors in comparison to other modules.

# Chapter Six: Conclusion and Future Work

## 6.1 Conclusion

NLG has been an area of immense interest among researchers during the last couple of decades. This area has witnessed a few lows and highs during its life span and has also observed integration of research works from different fields including linguistics, computer science, artificial intelligence, statistics, mathematics, and others. Researchers have proposed different paradigms of NLG for natural languages with reasonable success. This thesis explains the development of an Amharic sentence generator from Interlingua representation.

Interlingua is a representation which is unambiguous, universal and independent of both the source language and the target language. Generation will always be target language-specific, and likewise the analysis. Among the different Interlinguas, Universal Networking Language (UNL) is commonly chosen in view of various advantages over the other. Various works have been done so far for different languages using UNL representation. Moreover, UNL system has the potential to bridge the language barriers across the world with the development of  $2n$  components, while traditional approaches require  $n*(n-1)$  components, where  $n$  is the number of languages. UNL represents the information sentence by sentence in the form of UWs, UNL relations and UNL attributes. The concepts are expressed by UWs and UNL relations are used to specify the role of each word in a sentence. The subjective meanings intended by the author are expressed through UNL attributes. UNL system makes use of word dictionaries in the form of Language-UW lexicon of respective languages for its processing.

In this thesis, we have reviewed some research activities, more specifically the research in the area of UNL. The work on conversion of Brazilian Portuguese into UNL and vice-versa, DeConversion tools for Tamil language, 'HERMETO' system, French EnConverter and French DeConverter, UNL DeConverter for Chinese language, *Manati* DeConversion model, UNL-Nepali DeConverter, and Arabic MT system based on UNL have also been reviewed. Based on the reviewed literature we learn that UNL is capable of representing knowledge so that different languages can utilize it to produce a sensible sentence. In addition, UNL can be applied for knowledge management, multilingual search engine, language independent Universal Digital Library, etc.

Based on an intensive review of related research works, gaps identified through the review and also considering an enormous demand of efficient solutions in the area, we have proposed a framework for Amharic. Moreover, in this section, we have summarized the components used in the proposed architecture as well as their flow is put succinctly as follows. It first accepts UNL expression as an input and the UNL parser parses the input UNL expression to build a node-net from the input UNL expression. It also creates a data structure so that the other modules could access it and further modify the content according to the changes. Then, each node from the parsed node-net will be searched from the dictionary with the intention of outputting an equivalent Amharic root word with grammatical attributes attached. Moreover, the generator defines the word order in the generated sentence so that the output matches with a natural language sentence. Algorithms have been implemented for local reordering of simple sentences, compound sentences (UNL graph with a scope node) and special cases in reordering. The proposed system makes use of morphology rule base for Amharic language to handle attribute label resolution, relation label resolution, and noun, adjective, pronoun and verb morphology. Finally, in the Amharic Morphology Generator module, the function words are inserted to the morphed words.

The proposed system has been tested for 142 UNL expressions. These expressions are taken from UNDL foundation from ethics and science domain. The proposed system has achieved a fluency score of 3.0 (on a 4-point scale) and an adequacy score of 2.9 (on a 4-point scale). The proposed system has a word error rate of 28.94%. In consideration of results from other works accomplished on a similar area to the proposed work for different languages, the result is good. In another dimension, if we take the complexity of Amharic and the existing limited resources into consideration and the richness of the works done for other languages in NLP advancement, it is conspicuous that the results are very encouraging.

## **6.2 Contribution of the Research**

The main contributions of this thesis are summarized as follows.

### **Automatic Amharic sentence generator with native speaker acceptance**

The proposed system can generate an intelligible sentence from a UNL expression. Besides, the proposed work came up with a technique that could be further improved by combining machine learning approaches with the proposed rule based approach since preparing all the possible rules

for Amharic is difficult. Furthermore, the major parameters for inserting prepositions and postpositions in a sentence are identified and knowledgebase is also prepared for handling morphology generation as well as function word insertion.

### **A Framework for Amharic Sentence generation from knowledge base**

The proposed system also put the foundation for other Amharic NLP applications like MT, DS, CLIR and QA if Interlingua approach is used, the generation of Amharic sentence could be handled by the proposed system. In addition, it enables the Amharic readers to read the sentences in their language that are originally represented in UNL. Since UNL is able to represent knowledgebase, the proposed generator can construct a sentence in which the concepts are represented in a UNL expression. Moreover, this system will also motivate scholars working on different NLP applications to explore more about UNL or generally Interlingua because of the promising findings of our work.

## **6.3 Future work**

Amharic sentence generation from Interlingua representation proposed in this research work could be used in various applications of NLP. In addition, various research works use the methodology and findings of this research. We tried to illustrate that the proposed system addressed the problems stated and provides a proof with a prototype. Hence, this research work presents different area that can be further improved.

- The coverage and accuracy of the system can further be improved by expanding the UNL-to-Amharic word dictionary and enriching it with more semantic information.
- The rule base can further be improved linguistically to increase the accuracy of the system and to handle very large and complex sentences.
- The proposed system can further be tested on other Ethiopian languages by using corresponding language's rule base and lexicon, because of the closely knit in terms of kinship relations.

## References

- [1] Jesús Cardeñosa, Carolina Gallardo, and Edmundo Tovar, “Standardization of the Generation Process in a Multilingual Environment,” International Conference on the Convergence of Knowledge, Culture, Language and Information Technologies, Alexandria, Egypt, December 2 - 6, 2003.
- [2] Dale, Robert and Reiter, Ehud, “Building Natural Language Generation Systems,” Cambridge, U.K.: Cambridge University Press. ISBN 0-521-02451-X, 2000.
- [3] W. J. Hutchins and H. L. Somers, “An Introduction to Machine Translation,” (chapter1, 4, 17) (chapter 1 p.8) London Academic Press Limited, 1992.
- [4] Smriti Singh, Mrugank Dalal, Vishal Vachhani, Pushpak Bhattacharyya, and Om P. Damani, “Hindi Generation from Interlingua (UNL),” Indian Institute of Technology, Bombay, India, 2007.
- [5] Alex Waibel, Ahmed Badran, Alan W Black, Robert Frederking, Donna Gates, Alon Lavie, Lori Levin, Kevin Lenzo, Laura Mayfield Tomokiyo, Jurgen Reichert, Tanja Schultz, Dorcas Wallace , Monika Woszczyna, and Jing Zhang, “Speechalator: Two-Way Speech-To-Speech Translation on a Consumer PDA”, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA Cepstral, LLC, Multimodal Technologies Inc and Mobile Technologies Inc, 2003.
- [6] K. Shaalan, A. Abdel Monem, A. Rafea, and H. Baraka. “Syntactic Generation of Arabic in Interlingua-based Machine Translation Framework,” Third workshop on Computational Approaches to Arabic Script-based Languages (CAASL3), Machine Translation Summit XII, Ottawa, Ontario, Canada, August 26, 2009.
- [7] Azza Abdel Monem, Khaled Shaalan, Ahmed Rafea, and Hoda Baraka, “Generating Arabic text in multilingual speech-to-speech machine translation framework,” Machine Translation, pp. 205-258, December 2008.
- [8] Busemann S., “Issues in Generating Text from Interlingua Representations,” In Proceedings of the 1st International Workshop on UNL, other Interlinguas and their Applications, Las Palmas, Spain, 2002.

- [9] Bharati A., Chaitanya V., and Sangal R., “Natural Language Processing: A Paninian Perspective,” Prentice-Hall of India, New Delhi, 1994.
- [10] Siddiqui T., and Tiwary U. S., “Natural Language Processing and Information Retrieval,” Oxford University Press, New Delhi, 2008.
- [11] María Rosario, and Bautista Zambrana, “Natural Language Generation and Translation Technologies,” University of Malaga, Malaga, Spain.
- [12] Bonnie J. Dorr, E. Hovy, and L. Levin, “Machine Translation: Interlingual Methods”, Encyclopedia of Language and Linguistics. 2nd ed., Brown, Keith (ed.), 2004.
- [13] McDonald, David D., “Description directed control: Its implications for natural language generation,” Computers & Mathematics with Applications 9.1, 1983.
- [14] Nalawade A., “Natural language generation from Universal Networking Language,” Unpublished M. Tech. Thesis, IIT Bombay, Mumbai, 2007.
- [15] Elhadad M., and Robin J., “Controlling Content Realization with Functional Unification Grammars,” Aspects of Automated Natural Language Generation, pp. 89-104. Springer, Heidelberg, 1992.
- [16] Daniel Jurafsky, and James H. Martin, “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, And Speech Recognition,” 2nd Ed. New Jersey: Prentice Hall, 2009.
- [17] A. Lampert, “Interlingua in Machine Translation,” Technical Report, 2004.
- [18] W. J. Hutchins, “Machine Translation: A Brief History,” Concise History of the Language Sciences: from the Sumerians to the Cognitivists. E.F.K. Koerner and R.E. Asher (eds.), Pergamon, pp. 431-445, 1995.
- [19] H. Uchida and M. Zhu, “Interlingua for Multilingual Machine Translation,” in Proceedings of the Machine Translation Summit IV, Kobe, Japan, July 20-22, 1993.
- [20] UNDL Foundation 2010, <http://www.undl.org>, accessed 19 July 2015.
- [21] T. Witkam, “History and Heritage of the DLT (Distributed Language Translation) project,” Utrecht, The Netherlands: private publication, 2006.
- [22] Alansary S., “Interlingua-based Machine Translation Systems: UNL versus Other Interlinguas”, in Proceedings of the 11th International Conference on Language Engineering, Cairo, Egypt, 2011.

- [23] Christian Boitet, “A Rationale for Using UNL as an Interlingua and More in Various Domains,” First International Workshop on UNL, other Interlinguas and their Applications, 2005.
- [24] H. Uchida and M. Zhu, “UNL for Providing Knowledge Infrastructure,” in Proceedings of the Semantic Computing Workshop (SeC2005), Chiba, Japan, 2005.
- [25] Alansary S., Nagi M., and Adly N., “Towards a Language-Independent Universal Digital Library,” Proc. 2nd Int. Conf. on Universal Digital Library, Alexandria, Egypt, 1-10, 2006.
- [26] Mukerjee A, Achla M R, Kumar K, Goyal P, and Shukla P, “Universal Networking Language: a Tool for Language Independent Semantics?” Universal Network Language: Advances in Theory and Applications, Ed(s) Cardeñosa J, Gelbukh A, Tovar E, México, Research on Computing Science: pp. 145-154, 2003.
- [27] Bértoli L., Luz R. P., and Bastos R. C., “A web platform using UNL: CELTA’s Showcase. Universal Network Language: Advances in Theory and Applications,” Ed(s) Cardeñosa J, Gelbukh A, Tovar E, México, Research on Computing Science: 276-285, 2005.
- [28] Surve M., Singh S., Kagathara S., Venkatasivaramasastry K., Dubey S., Rane G., Saraswati J., Badodekar S., Iyer A., Almeida A., Nikam R., Perez C. G., and Bhattacharyya P., “AgroExplorer: a Meaning Based Multilingual Search Engine,” Proc. Int. Conf. on Digital Libraries, Delhi, India: 1-13, 2004.
- [29] Ramamritham K., Bahuman A., and Duttagupta S., “aAqua: a Database-Backended Multilingual, Multimedia Community Forum,” Proc. Int. Conf. on Management of Data, Chicago, USA, pp. 784-786, 2006.
- [30] CSA, The 2011 Ethiopia Statistical Abstract, Central Statistical Agency, Addis Ababa, Ethiopia, 2011.
- [31] Isenberg, C.W., “Grammar of the Amharic Language,” London, Church Missionary Soc., 1842.
- [32] Shukla P., Mukherjee A., and Raina A., “Towards a language independent encoding of documents a novel approach to multilingual question answering,” Proc. 1st Int. Workshop on Natural Language Understanding and Cognitive Science, Porto, Portugal, pp. 116-125, 2004.
- [33] Baye Yimam, “Yä Amarəña Säwasəw,” E.M.P.D.A, Addis Ababa, 1987.

- [34] Wolf Leslau, "Reference Grammar of Amharic," Wiesbaden: Otto Harrassowitz, 1995.
- [35] Markos Kassa Gobena, "Implementing an Open Source Amharic Resource Grammar in GF," Chalmers University of Technology, Göteborg, 2010.
- [36] Aynadis Temesgen Gebru, "Design and Development of Amharic Grammar Checker," Unpublished MSc Thesis, Addis Ababa University, Ethiopia, 2013.
- [37] Amare Getahun, "Issues of Time and Place Adverbs in Amharic," *African Languages and Cultures* 8, pp. 123-136. 1995.
- [38] Choudhary and Narayan Kumar, "Developing a Computational Framework for the Verb Morphology of Great Andamanese", Centre for Linguistics, JNU, 2006.
- [39] Shambhavi. B. R, Ramakanth Kumar P, Srividya K, Jyothi B J, Spoorti Kundargi, and Varsha Shastri G, "Kannada Morphological Analyser and Generator Using Trie," *International Journal of Computer Science and Network Security (IJCSNS)*, Vol.11 No.1, January 2011.
- [40] Keshari B, and Bista S K, "UNL Nepali Deconverter," *Proc. 3rd International CALIBER*, Cochin, 70-76, 2005.
- [41] Martins R T, Hasegawa R, Graças M, and Nunes V, "Hermeto: A NL–UNL Enconverting Environment," *Universal Network Language: Advances in Theory and Applications*, Ed(s) Cardeñosa J, Gelbukh A, Tovar E, México, *Research on Computing Science*: 254-260, 2005.
- [42] Dhanabalan T., and Geetha T. V., "UNL DeConverter for Tamil," *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies*, Alexandria, Egypt, pp. 1-6, 2003.
- [43] Tanaka H, Ishizaki S, Uehara A, and Uchida H, "Research and Development of Cooperation Project on a Machine Translation System for Japan and its Neighboring Countries," *Proc. of 2nd MT Summit*, Munich, pp. 146-151, 1989.
- [44] Blanc É, "About and around the French Enconverter and the French Deconverter," *Universal Network Language: Advances in Theory and Applications*, Ed(s) Cardeñosa J, Gelbukh A, Tovar E, México, *Research on Computing Science*, pp. 157-166, 2005.
- [45] Daoud D. M., "Arabic Generation in the Framework of the Universal Networking Language," *Universal Network Language: Advances in Theory and Applications*, Ed(s)

- Cardenosa J, Gelbukh A, Tovar E, México, *Research on Computing Science*: 195-209, 2005.
- [46] Collins M., “Head-driven statistical models for natural language parsing,” *Computational linguistics*, 29(4), pp.589-637, 2003.
- [47] Klein D., and Manning C.D., “Accurate unlexicalized parsing,” In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pp. 423-430, Association for Computational Linguistics, 2003, July.
- [48] Balkan L, “Test Suites: Some Issues in their use and Design,” *Proc. Int. Conf. on Machine Translation: ten years on*, England, pp. 214-222, 1998.
- [49] Slype V, “Critical Study of Methods for Evaluating the Quality of Machine Translation,” *Technical Report*, European Commission, Brussels, pp. 1-185, 1979.
- [50] Ma Xiaoyi, and Christopher Cieri, “Corpus support for machine translation at LDC.” *Proceedings of LREC*, 2006.

# Annexes

## Annex A: UNL Relations

UNL Relation	Description	Constituent Element	Examples
1. agt	defines a thing that initiates an action	Agent, intransitive verbs  (intransitive verb semantically have an agent subject) like ተኛ፣ አንኮራፋ፣ ሳለ፣ ሮጠ  ...	agt(slept, John) ጆን ተኛ  agt(killed, John) ጆን ገደለ  agt(arrival, John) የጆን መምጣት  agt(play, Shakespeare) የሼክስፒየር ቴያትር
2. and	defines a partner to have conjunctive relation	Conjunction	and(quickly, easily) በቀላል እና በፍጥነት  and(dance(agt>person),sing(agt>person)) መዝፈን እና መጨፈር
3. aoj	defines a	Stative verbs like እምነት፣ መረዳት፣ ማወቅ፣ መጥላት፣	aoj(believes, John) ጆን ያምናል

	thing that is in a state or has an attribute	ማፍቀር፣ መውደድ፣ ማካተት...	<p>aoj(knows, John) ጆን ያውቃል</p> <p>aoj(loves, John) ጆን ይወዳል</p>
		aoj (general attribute)	<p>aoj(sad, John) ጆን አዘነ</p> <p>aoj(sad, John) ጆን ያዘነ ይመስላል</p>
4. bas	defines a thing used as the basis (standard)of comparison	Basis	<p>bas( more(aoj&gt;thing,bas&gt;thing), 7) ከ7 ይበልጣል</p> <p>bas(more(icl&gt;how,bas&gt;thing), Jack(iof&gt;person)) ከጆን ይሻላል</p>
5. ben	defines an indirectly related beneficiary or victim of an event or state	Beneficiary	<p>ben(give(agt&gt;thing, gol&gt;thing,obj&gt;thing), country(icl&gt;region)) ለሀገር መስጠት</p> <p>ben(good(aoj&gt;thing), John(iof&gt;person)) ለጆን ጥሩ</p>

6. cag	Defines a thing not in focus that initiates an implicit event that is done in parallel	Co-agent	cag(walk(agt>volitional thing), John(iof>person)) ከጆን ጋር መራመድ  cag(live(agt>volitional thing), aunt(icl>person)) ከአክስት ጋር መኖር
7. cao	Defines a thing not in focus that is in a parallel state	Co-thing with attribute	cao(exist(aoj>thing),you) ከአንተ ጋር መሆን
8. cnt	defines the content of a concept	Content	cnt( Internet(icl>communication network), amalgamation(icl>harmony)) ኢንተርኔት፡- ውህድ  cnt(language generator, deconverter.@double_quote) የቋንቋ ሞተር፡- “ዲኮንቨርተር”
9. cob	defines a thing that is directly affected by an implicit event done in parallel or an implicit state in parallel	Affected co-thing	cob(die(obj>livingthing), Mary(iof>person)) ከሜሪ ጋር አብሮ ሞት

10. con	defines a non-focused event or state that conditions a focused event or state	Condition	aoj:01(tired(aoj>thing),you) con(go(icl>move(agt>thing, gol>place,src>place)), :01) አንተ ከደከመህ እሄዳለሁ
11. coo	defines a concurrent event or state for a focused event or state	While	coo(work, talk) እየአወራሁ ሰራሁ
12. dur	defines a period of time during which an event occurs or a state exists	During	dur(work, meet) በሰብሰባ ሰዓት ሰራ
13. equ	defines an equivalent concept	Equivalent	equ(deconverter, language generator.@parenthesis) ዲኮንቨርተር (የቋንቋሞተር)
14. fmt	defines a range between two things	Range/from-to	fmt(z(icl>letter), a(icl>letter)) ከኤ እስከ ዜድ  fmt(New York(iof>city), Osaka(iof>city)) ከአሳካ እስከ ኒውዮርክ
15. frm	defines an	Origin	frm(visitor(icl>person), Japan(iof>country))

	initial state of a thing or a thing initially associated with the focused thing		ከጃፓን ጎብኚ
16. gol	defines a final state of object or a thing finally associated with the object of an event	Final state of verbs of change like ሰጠ፣ ላከ	gol(gave, Mary) ለሜሪ ሰጠ  gol(sent, Mary) ለሜሪ ላከ
17. icl	defines an upper concept or a more general concept	Included/a kind of	icl(bird(icl>animal), animal(icl>livingthing)) ወፍ የእንሰሳ አይነት
18. ins	defines an instrument to carry out an event	Instrument	ins(look(agt>thing,obj>thing), telescope(icl>optical instrument)) በአጉሊ መነጻር ተመልከት  ins(write(agt>thing,obj>thing), pencil(icl>stationery)) በእርሳስ ጻፍ
19. int	defines all common instances to have with a partner concept	Intersection	int(tableware(icl>tool), cookware(icl>tool))
20. iof	defines a class concept that an instance belongs to	An instance of	iof(Tokyo(iof>city), Japan) ቶኪዮ ጃፓን ውስጥ ያለ ከተማ

21. man	defines a way to carry out an event or the characteristics of a state	Manner	man(move(agt<thing, gol>place,src>place), quickly) በፍጥነት ተንቀሳቀስ  man(visit(agt>thing, obj>thing), often) በተደጋጋሚ ጎብኝ
22. met	defines a means to carry out an event	Method/ means	met(solve(icl>resolve(agt>thing,obj>thing)), dynamics(icl>science)) በዳይናሚክስ ፈታ  met(separate(agt>thing,obj>thing,src>thing), cut(agt>thing,obj>thing, opl>thing)) በመቁረጥ ለየ
23. mod	defines a thing that restricts a focused thing	Modification	mod(story(icl>tale), whole(mod<thing)) ሁሉንም ታሪክ  mod(plan(icl>idea),master(mod<thing)) መሪው እቅድ
24. nam	defines a name of a thing	Name	nam(son(icl>relative), Hikari) ልጅ ሂካሪ
25. obj	defines a thing in focus that is directly	Un-Accusative verbs like ሞተ፣ ወደቀ፣ ቀለጠ ...	obj(died, John) ጆን ሞተ

affected by an event or state		obj(melts,snow) በረዶ ቀለጠ
	obj (direct object)* Accusative	obj(killed, Mary) ሜሪን ገደለ  obj(knows, Mary) ሜሪን ያውቃታል  obj(loves, Mary) ሜሪን ይወዳታል
	obj (indirect object of mono-transitive verbs like ተመረኮዘ፣ አመነ፣ ሳቀ	obj(depends, Mary) በሜሪ ላይ ተመረኮዘ  obj(believes, Mary) በሜሪ አመነ
	object	obj(construction, building) የህንጻ ግንባታ  obj(interest, Physics) የፊዚክስ ፍላጎት  obj(visit, London) የለንደን ጉብኝት

26. opl	defines a place in focus affected by an event	Affected place	opl(pat(icl>touch(agt>thing, obj>thing, opl>thing)), shoulder(pof>trunk)) ትከሻ ላይ ጉብ  opl(cut(agt>thing, obj>thing, opl>thing), middle(icl>place)) መሃል ላይ ቆረጠ
27. or	defines a partner to have disjunctive relation to	Disjunction	or(leave(agt>thing, obj>place), stay(icl>remain(agt>thing))) ቆይ ወይም ሂድ  or(blue(icl>color), red(icl>color)) ቀይ ወይስ ሰማያዊ
28. per	defines a basis or unit of proportion, rate or distribution	Proportion/ rate/ distribution	per(hour(icl>period), day(icl>period)) በቀን ሰዓት
29. plc	defines a place where an event occurs, or a state is true, or a thing exists	Place	plc(made, Italy) በጣልያን ተሰራ ጣልያን ውስጥ/ላይ ተሰራ
30. plf	The place where an event begins or a state becomes true	Initial place of verbs of motion like ሄደ፣ ተጓዘ፣ ተራመደ፣ መጣ...	plf(came, German) ከጀርመን መጣ

31. plt	defines a place where an event ends or a state that becomes false	Final place	plt(travel(agt>volitional thing), Panama (iof>city)) ወደ ፓናማ ተጓዘ
32. pof	indicate a concept of which a focused thing is a part	Part of	pof(preamble(icl>information), document(icl>information)) የሰነድ መግቢያ  of(initial(icl>letter), machine translation) የሚገኝ ትርጓሜ መነሻ
33. pos	defines the possessor of a thing	Possessor	pos(dog(icl>aminal), John(iof>person)) የጆን ውሻ  pos(book(icl>document),I) የእኔ መጽሕፍ
34. ptn	defines an indispensable non-focused initiator of an action	Partner	ptn(competete(agt>thing, ptn>thing), John(iof>person)) ከጆን ጋር ጨረሰ  ptn(share(icl>divide(agt>thing, obj>thing)), poor(icl>person)) ከድሃ ጋር ተካፈለ
35. pur	defines the purpose or objective of an agent of an event or	Purpose	pur(work(icl>do),money) ለገንዘብ ይሰራል

	the purpose of a thing that exists		pur(budget(icl>expense), research(icl>study)) ለምርምር ቢጀት
36. qua	defines the quantity of a thing or unit	Quantity	qua(cup(icl>tableware),2)) 2 ሲኒ  qua(coffee(icl>beverage), cup(icl>tableware)) ሲኒ ቡና  qua(kilogram(icl>unit), many(qua<thing)) ብዙ ኪሎ  qua(dog(icl>animal), 2) 2 ውሻዎች
37. rsn	defines a reason why an event or a state happens	Reason	rsn(go(icl>do), illness(icl>thing)) በህመም ምክንያት ሄደ  rsn(known(aoj>thing), beauty(icl>abstract thing)) በቁንጅና ምክንያት ታወቀ
38. scn	defines a scene where an event occurs, or state is	Scene	scn(appear(icl>occur), program(icl>thing))

	true, or a thing exists		መርጎግብር ላይ ኖረ
39. seq	defines a prior event or state of a focused event or state	After	seq(left, worked) ሰርቶ(Gerundive)ሄደ
40. shd	defines a number, a mark or a thing that shows the position of a sentence, a paragraph or a chapter in a document or a book	Sentence head	shd(relation(icl>sate), chapter(pof>book)) mod(chapter(pof>book),2) ምዕራፍ 2:- ግንኙነት
41. src	defines the initial state of an object or thing initially associated with the object of an event	Initial state of verbs of change like ወሰደ፣ ተቀበለ፣ ተለወጠ...	src(change(icl>occur), red(aoj>thing)) ከቀይ ተለወጠ
42. tim	defines the time an event occurs or a state is true	When	tim(came,yesterday) ትላንት መጣ
43. tmf	defines the time an event starts or a state becomes true	Since when	tmf(worked, early) ከጥዋት ጀምሮ ሰራ
44. tmt	defines the time an event ends or a state becomes false	Until when	tmt(worked, late) እስከ ምሽት ሰራ

45. to	defines a final state of a thing or a final thing (destination) associated with the focused thing	Destination	to(train(icl>thing), London(icl>city)) ወደ ለንደን ባቡር
46. via	Defines an intermediate place or state of an event	An intermediate place or state	via(go(icl>do), New York(icl>place)) በኒውዮርክ አድርጎ ሄደ

## Annex B: Priority Matrix of UNL Relations for Amharic

H i g h	a g t	a c t	s h d	m a n	a a j	o b j	p u r	f m r	f r m	r s s	b a s	t i m	p l c	b e n	g o l	i n s	q u a	m o d	e q u	v i a	o f	p a g	c o b	c a o	s e q	p t n	n a m	m e l	p l o	o p l	s c n	t m f	t m t	p l t	i o f	i n t	d u r	c o n	c o r	p e r	s e q					
a g t	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
c n t	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
s h d	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
o r	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
m a n	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
a n d	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
a o j	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
o b j	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

H  
i  
g  
h

	a	c	s	m	a	a	o	p	f	f	r	b	t	p	b	g	i	q	m	e	v		p	c	c	c	s	p	n	m	p	p	o	s	t	t	p	i	i	d	c	c	p	s	e			
	g	n	h	o	n	o	b	u	m	r	s	a	i	l	e	o	n	u	o	q	i	t	o	a	o	a	e	t	a	e	l	o	p	c	m	m	l	o	n	u	o	o	r	c	u			
P u r	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
f m t	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
f r m	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
r s n	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
b a s	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
t i m	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
p l c	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
b e n	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

H  
i  
g  
h

	a	c	s		m	a	a	o	p	f	f	r	b	t	p	b	g	i	q	m	e	v		p	c	c	c	s	p	n	m	p	p	o	s	t	t	p	i	i	d	c	c	p	s	e					
g	o	l	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
i	n	s	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
q	u	a	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
m	o	d	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
e	q	u	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
v	i	a	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
t	o		R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
p	o	f	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

H  
i  
g  
h

	a	c	s	m	a	a	o	p	f	f	r	b	t	p	b	g	i	q	m	e	v	p	c	c	c	s	p	n	m	p	p	o	s	t	t	p	i	i	d	c	c	p	s	e														
	g	n	h	o	a	a	o	b	u	m	r	s	a	i	l	e	o	n	u	o	q	i	t	o	a	o	a	e	t	a	e	l	f	s	l	n	f	t	t	p	i	i	d	c	c	p	s	e										
g	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R						
o	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R					
a	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R				
s	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R			
p	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R			
n	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
m	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
p	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

H  
i  
g  
h

	a	c	s		m	a	a	o	p	f	f	r	b	t	p	b	g	i	q	m	e	v		p	c	c	c	s	p	n	m	p	p	o	s	t	t	p	i	i	d	c	c	p	s	e		
	g	n	h	o	a	a	o	b	u	m	r	s	a	i	l	e	o	n	u	o	q	i	t	o	a	o	a	e	t	a	e	l	o	p	c	m	m	l	o	n	u	o	o	r	c	u		
P o s	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	L	
o p l	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	
s c n	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	L	
t m f	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	
t m t	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L	L	
p l t	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L	L
i o f	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L
i n t	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	L	L	L	L	L

H	a	c	s	m	a	a	o	p	f	f	r	b	t	p	b	g	i	q	m	e	v	p	c	c	c	s	p	n	m	p	p	o	s	t	t	p	i	i	d	c	c	p	s	e												
i	g	n	h	o	a	n	o	b	u	m	r	s	a	i	l	e	o	n	u	o	q	i	t	o	a	o	a	e	t	a	e	l	o	p	c	m	t	l	o	n	u	o	o	e	r	q										
g	t	t	d	r	n	d	j	j	r	t	m	n	s	m	c	n	l	s	a	d	u	a	o	f	g	b	o	q	n	m	t	f	s	l	n	f	t	t	f	t	r	n	o	r	c	u										
	d																																																							
	u	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L							
	r																																																							
	c	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L							
	o																																																							
	n	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L						
	c																																																							
	o	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L						
	o																																																							
	o	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L					
	p																																																							
	e	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L			
	r																																																							
	r	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L			
	s																																																							
	r	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L		
	c																																																							
	u	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L	
L	e																																																							
o	q	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	L	L	L	L	L
w	u																																																							

