SUPERVISED AUTOMATIC SENTENCE PARSING FOR HADIYAA LANGUAGE

DEREJE YAKOB JELETIE

A Thesis Submitted to

The Department of Computing

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Information Systems

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

September, 2017

Declaration

I, the undersigned, declare that this thesis entitled “Supervised Automatic Sentence Parsing for Hadiyya Language” is my original work and has not been submitted for a degree of MSc. and publication in any other University. In addition, all the source of material used for the thesis has been accordingly acknowledged.

_____

Dereje Yakob

This thesis has been submitted for examination with my approval as university advisor.

_____

Dr. Shimelis Assefa (Asso. Professor)

[Advisor]

September, 2017

Acknowledgement

First, I would like to thank almighty God Father, Jesus and Holy Spirit for helping me to finalize this thesis. My deepest heartfelt gratitude also goes to my advisor Dr. Shimelis Assefa, (Associate Professor of Denver University), for his remarkable advice and critical comments from the very start of the research work to the end.

I would like to thank Dr. Samuel Handamo lecturer at AAU language department of foreign language and leterature for his linguistic advice and provision of necessary Hadiya language resources and Mr. Zereyakob G/sillasse (PhD. Student at AAU in Educational Policy and Leadership Department) for his basic constructive support on document commenting throughout the document preparation.

My special thanks goes to my mother W/o Almaz Aossa, for her support with prayer and to my father, all brothers and sisters for strengthen with suggestion, sharing idea throughout my education time.

Finally, last not least, I would like to thank all my friends for their supportive idea valuable for conducting the research.

Abstract

The Syntactic Processing (parsing) is the method for understanding how words and phrases are combined to synthesize a sentence whether it fits to the given language grammar structure. Parsing is an essential intermediate phase for the building language semantic analysis systems. This research is mainly concerned with developing prototype to automatic sentence parsing for Hadiyya Language using hybrid approach. Thus, to conduct this research Rule-Based Learning Approach with Chart parsing algorithm combined with automatic feature construction algorithms (Maximum Entropy model) used to realize the parser.

In this thesis a modified version of the Chart parsing algorithm was applied to a corpus that was manually annotated with a skeletal syntactic bracketing which confirms to the Penn Treebank POS Tags. The Hadiyya language parsing system prototype is evaluated with PARSEVAL metrics, which computed by Precision, Recall and F-measure. The experment conducted on the total corpus of 549 sentences colections. The obtained results were 73.0 % precision and Recall value of 64.3 % on test data set. The higher precision value when we compared with recall showed us the parser is effective in syntactic analysis.

Key Words: ***Automatic Sentence Parsing, Sentence Parsing, Hadiyya Language***

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviations	Description
ADVP	Adverb Phrase
CD	Cardinal Number
JJ	Adjective
NN	Noun
NP	Noun Phrase
PCFG	Probablistic ContextFree Grammar
PP	Prepositional Phrase
POS	Parts of Speech
S	Declarative sentence
VP	Verb Phrase
VR	Verb
VN	Verbal Noun
WFR	Word Formation Rule

Definitions of Terms

Natural language- Human Spoken Language

Morphology— knowledge of the meaningful components of words

Parsing- is a method to perform syntactic analysis of a sentence.

Phonetics and Phonology— knowledge about linguistic sounds

Syntax- knowledge of the structural relationships between words

Treebanks- are language resources in which the syntactic structure of a corpus of sentences has been annotated, usually by hand.

CHAPTER ONE

BACKGROUND OF THE STUDY

1.1. Introduction

Automatic sentence parsing technology is concerns how words can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of what other phrases for the natural language. The study of Natural Language Parsing of written human natural language has been research area providing research workers and practitioners with a range of implementation and investigative techniques starting from 1950's is still continuing as essential, both at the programming and the analytic levels. Parsing is a formally defined process, which is concerned with the abstract mechanisms for applying structure-determining rules to input strings (Jones Karen, 1985) (Alexander Clark, 2010).

The natural language processing work is mostly concerned with the basic unit of meaning analysis of a sentence. The sentence expresses a proposition, an idea, or a thought, and says something about some real or imaginary world. Extracting the meaning from a sentence is thus a key issue. Sentences are not, however, just linear sequences of words, and so it is widely recognized that to carry out this task requires an analysis of each sentence, which determines its structure in one way or another. In NLP approaches based on generative linguistics, it involves the determination of the syntactic or grammatical structure of each sentence. This area is probably the most well established in the field of NLP, enabling the authors to provide an inventory of basic concepts in parsing, followed by a detailed catalog of parsing techniques that have been explored in the literature (Alexander Clark, 2010). In this thesis a range of techniques are presented that can be used to identify and achieve this end.

According to (Daniel Jurafsky, 2009), Parsing is one of the best-understood phases of compilation. From a set of syntactic rules, it is possible to automatically construct parsers, which will make sure that a source program obeys the syntactic structure defined by these syntactic rules. We shall review several different parsing techniques and algorithms for generating a parser from a given grammar.

The output from the parser is a tree, which represents the syntactic structure inherent in the source program. In many ways, this tree structure is closely related to the parsing diagrams.

Currently, many researches have been conducted for natural language processing specifically on parsing technology for guiding machine translation. The study contributes for the speakers of the language to efficiently utilize the language without disrupting structural rule of the spoken language to be able to represent with sentences to the natural language processing machine system. For instance, the nations that gain advantage from the research related to parsing are English and Indian (Hindi, Bangla and Telgu and others), from inland, Amhara, Oromo, Wolayta, Tigre with known published studies, but the Hadiyya speakers don't have the parser that enables to process texts according to its grammar rule. The aim of this study is to develop Automatic Sentence Parsing (Syntactic Processing) prototype of Hadiyya, a Cushitic language spoken in the south west of Ethiopia. For this, the study will use supervised learning technique and modified rule based and chart parsing algorithms to realize the parser.

Hadiyya language is not endangered from eighty (80) Nations Nationality of Ethiopia. Thus, currently Hadiyya nations geographically located in the Ethiopian Southern Nations Nationalities and Peoples Region (SNNPR) Hadiyya Zone. Hadiyya people have their own language, historical origin, development and located in different regions of Ethiopia. The central town Wachamo/Hossana/ is located in the north center of the Hadiyya zone at a distance of 232 km south of Addis Ababa and 160 km west of Hawassa town. The Hadiyya language is used by the native people which is more than one million and neighboring people also communicate as their second language. The Hadiyya zone neighboring people are different nationalities surrounding this zone, (to the North by Silti and Gurage, to the south by Wolayitta, to the southeast by Kambaata, by Tambaaro to the south west and in the west by Omo river which separates it from Oromia region and the Yem Special Woreda) (Tadesse, 2015). In addition, the Hadiyya language speakers /users / are also available in the Benishangul Gumuz Regional state at Pawe Woreda and Ormiya regional state around illubabor.

Syntactic analysis (parsing) is the process of analyzing and determining the grammatical structure of a sentence with respect to a given formal grammar. According to the surveyor (Monika T. MakwanaȦ, 2015), 'Parsing' is the term used to describe the process of automatically building syntactic analyses of a sentence in terms of a given grammar and lexicon. The resulting syntactic analyses can be used as input to a process of semantic parsing, (or perhaps phonological

interpretation, where aspects of this, like prosody, are sensitive to syntactic structure). Occasionally, 'parsing' is also used to include both syntactic and semantic analysis (Pulman, 1991).

There are several important properties that a parsing algorithm should have if it is to be practically useful. It should be 'sound' with respect to a given grammar and lexicon; that is, it should not assign to an input sentence analyses which cannot arise from the grammar in question. It should also be 'complete'; that it, it should assign to an input sentence all the analyses it can have with respect to the current grammar and lexicon. Ideally, the algorithm should also be efficient, entailing the minimum of computational work consistent with fulfilling the first two requirements, and 'robust': behaving in a reasonably sensible way when presented with a sentence that is unable to fully analyze successfully [(Pulman, 1991), (Alexander Clark, 2010)].

Natural language grammars are combination of human spoken sentences. Which is formed using different category of parts of speech, that it needs rearrangement. It is advantageous that scientifically well formulated machine translation technique is applied to develop efficient parser. To realize sentence parsing customizing already existing algorithm gives a chance to integrate different models of machine training techniques. Natural language text is generally, not made up of the short, neat, well-formed, and well-delimited sentences.

Machine learning techniques for parsing a sentence can be supervised, unsupervised method or a hybrid. The supervised method uses labeled data and human controls it. Unsupervised learning approach is definitely controversial to supervised machine learning approach, because it uses unlabeled data for training a parser. Thus, the ability of self-training by unsupervised learning approach having problem with complex pattern. The hybrid approach integrate important features from both supervised and unsupervised machine learning techniques.

Automatic parsing of sentence is important to avoid manual computation cost of sentence structural arrangement. There are a number of parsing algorithms for parsing or syntactic analysis of text / sentence of Natural Language. Some of the algorithms are top-down, bottom-up, chart and others. For this study, the chart parsing algorithm is used with little modification for it is an efficient algorithm. Chart parsing also gives a degree of robustness in the case of failure to find a complete parse, because either the input is ungrammatical or (what amounts to the same thing from the point of view of the parser) is not within the coverage of the grammar. Even if the overall parse fails, an algorithm will result in all well-formed sub-constituents that are present in the input being found. From these, different types of heuristic strategy can be used to do something useful with them [9].

Syntax based parsing of sentence is base for complete analysis of text based human language. According to the Handbook of Natural Language processing, traditionally processing a Natural Language tended to view the process of language analysis as being decomposable into a number of stages, mirroring the theoretical linguistic distinctions drawn between syntax, semantics and pragmatics. The simple view is that the sentences of a text are first analyzed in terms of their syntax. This provides an order and structure that is more amenable to an analysis in terms of semantics or literal meaning and this is followed by a stage of pragmatic analysis whereby the meaning of utterance or text in context is determined (Damerau, 2010).

In order to minimize the complexity of analysis to natural languages to capture speaker's intended meaning for the written, text the language processing needs to be decomposed well-known in to lexical analysis steps. As Damerau, N. and his colleagues' states that: Natural language text is generally not made up of the short, neat, well-formed, and well-delimited sentences. For example, in textbooks for languages such as Chinese, Japanese, or Thai, which do not share the apparently easy space-delimited tokenization that might believe to be a property of languages like English, the ability to address issues of tokenization is essential to get the ground at all. Therefore, the stage of analysis in processing natural language generally follows tokenization, lexical analysis, syntactic analysis, semantic analysis and pragmatic analysis.

An automatic parsing of natural language sentence is done by recognizing set of coded tags each corresponding to a respective part-of-speech of a word in a sentence of the language model. Building a small corpus of sentence dependency structures by annotating with the corresponding coded tags of the word in a set of sentences having respective exemplary dependency structures to obtain a set of sequences of coded tags representing the dependency structures. From the corpus of sentence tagging the words of an input sentence to a corresponding coded tags required to obtain a sequence of coded tags to derive there form the most probable dependency structure for the input sentence. It is possible to generate a parse tree of the sentence from the derived dependency structure and the words of the input sentence. The sentence parsing of a natural language involves resolving the sentence into its component parts and determining the structural relationship amongst the words from which the sentence is constructed. (United States of America Patent No. 5,930,746, 1999)

The natural language has its own infinite number of linear representation ("sentence") that need to be processed and structured according to the grammar of the language. Grammatical representation summarizes systematically an infinite set of structure for an object of a certain class. Thus, there

are several reasons to perform a structural processing known as parsing (Dick Grune, 2008). In book of parsing techniques there are three main reasons stated. The first reason is the fact that the obtained structure helps us to process the object further. When we know that a certain segment of a sentence is the subject, that information helps us in understanding or translating the sentence. Once the structure of a document has been brought to the surface, it can be converted more easily.
A second reason is related to the fact that the grammar in a sense represents our understanding of the observed sentences: the better a grammar we can give for the movements of bees, the deeper our understanding is of them.
A third lies in the completion of missing information that parsers, and especially error-repairing parsers, can provide. Given a reasonable grammar of the language, an error-repairing parser can suggest possible word classes for missing or unknown words on clay tablets.

In this time, there are many application area of natural language sentence parsing technology. Parsers are already being used extensively in a number of disciplines. In computer science (for compiler construction, database interfaces, self-describing databases, artificial intelligence), in linguistics (for text analysis, corpora analysis, machine translation, textual analysis of biblical texts), in document preparation and conversion, in typesetting chemical formulae and in chromosome recognition, to name a fews. (Damerau, 2010).

An automatic parsing for natural languages has been mostly rule-based. A pre-requisite of a rule-based system is a set of language dependent rules written and refined by linguists over many years. For a rule-based parsing machine to work, the rules of the particular language have to be identified and incorporated into the system. Writing grammar rules for rule-based parsing machines is a daunting task that only computational linguists can effectively perform. In devising such rule based systems, it has been necessary to formalize the grammar in systems such as the generalized context-free grammar formalism and the tree-adjoining grammar formalism (United States of America Patent No. 5,930,746, 1999).
This research has many contributions for the users of Hadiyya language and researchers upon the completion. To mention, some benefits are stated as follows: the main contribution of this research is to develop Hadiyya language sentence parser thus gives the language users to have their own language parser that enable to process texts based on the syntax of the sentence. The integration of machine learning algorithm and natural language processing method (like supervised learning and syntactic processing, respectively) enable to create natural language understanding for automatic

natural language processing systems. Mainly, this study is concerned to build a parser for Hadiyya language to setup ease in communication and use between this natural language user nations' and computer systems.

Hadiyya language grammar is gaining the advantage of being modelled or building syntactic structure that will give the opportunity for further study to succeeded for computational linguistic technology implementation. In order to design and develop the syntactic structure, the algorithms and models should be tested and customized as this is the main objective of this study.

For the researchers, this study contributes to perform further analysis of the language and as a base of sematic and discourse analysis. In educational linguistic institutions, this is directly being applied for Hadiyya language teaching for sentence processing.

This study is bounded to syntactic parsing of the Hadiyya language. Thus, semantic parsing for this language is not included under the scope.

1.2. Statement of the Problem

In the globalization age of information, every language of human being (the nations) has a significant contribution for communication of the message encoded within the computer, decoded for supporting application of decision-making. In order to compute the human language, we need to represent to a machine by using transcript following the structural arrangement of linguistic communication argument (grammar). The Natural Language Processing technique syntactic processing (parsing) is required for the language in order to do analysis on syntax or sentence structure of the given society. Some research has been conducted on Hadiyya language in linguistic departments such as predicate conjoining in Hadiyya: Head-driven Phrase Structure Grammar (HPSG) (Sim, 1989) and Documentation and Description of Hadiyya (A Highland East Cushitic Language of Ethiopia) a dissertation (Tadesse, 2015). However, the syntactic parser for Hadiyya language is not yet developed and the research was not conducted.

The automatic parsing avoiding preliminary manual adjustment and training on the annotated corpora is of great theoretical and practical interest. The resulting grammar rules can support the processes of language acquisition by people and the general structure of language, providing preliminary processing of texts for syntactic marking of large corpora and in the long term, ensures analysis of texts for natural language processing (Potemkin S, 1998). Thus, it is necessary to design the new algorithm for the languages that needs to have its own parsing tools and it is important to

be used as algorithm integration module to real application of syntax processing like spelling and grammar checker and others.

In Ethiopia, more than 80 languages are spoken and from this languages Hadiya Language is the one that used as a medium of instruction and taught as one subject starting from elementary unto college level (Dumessa, 2011). Non-native speakers and non-linguistic persons need help from word processing systems of natural language for task accomplishment. For example, writing clear and grammatically structured texts whenever needed. In addition, the meaning of the message encoded to the text for the readers never misguided. An automatic sentence parser can support compiling and transmitting the intended message from non-experts to the Hadiyya language users. Developing an automatic sentence parser for parsing machine give capability of accuracy, of processing sentences written in various styles, robust, easy for non-linguists to build, fine-tune and maintain, and fast.

Furthermore, designing an automatic parser is necessary in order to parse large corpora for computer enabled linguistic processing on real time processing systems.

Each language has some difference with structural representation of alphabet occurrence in word and phrases even though there is similar language group and some common words. Sentences are not, however, just linear sequences of words, and so it is widely recognized that to carry out this task requires an analysis of each sentence, which determines its structure in one way or another. In NLP approaches based on generative linguistics, this is generally taken to involve the determining of the syntactic or grammatical structure of each sentence (Damerau, 2010).

1.3. Objectives of the Study

1.3.1. General Objective

The main objective of this research is to examine a model that fits for parsing the language and designing automatic sentence parser of the Hadiyya language.

1.3.2. Specific Objectives

In order to accomplish the main objective of this study here is the lists of specific objectives.

- To review parts of speech (POS) of the Hadiyya language.
- To conduct a review on a grammatical structure for Hadiyya language (like, word, phrase and sentence) and adopt to the computer.
- To design an appropriate knowledge base for the parser to be developed.
- To study and review the existing and implemented models of parsers of the other languages.

- To select and customize a parsing algorithm that will result best efficient parsing for the desired language.
- To test the prototype that developed in this research as a parser for the language to measure its performance.

1.4 Research Questions

The following statements are focus questions that was addressed at the end of the study.

- Which of the existing parsing algorithms work better for Hadiyya language?
- What are the benefits of customizing of existing algorithm to develop syntactic parsing prototype for Hadiyya language?

1.5. Scope and Limitation of the Study

1.5.1. Scope of the Study

Because of the Hadiyya Nation lack of its sentence parsing prototype, this study is conducted only for the Hadiyya language, which is selected for this research. In addition, it is bounded to syntactic processing (parsing) not including sematic parsing of Natural Language Processing (NLP) using a hybrid machine learning approach.

1.5.2. Limitation of the Study

This study is conducted for only Hadiyya language and it does not include others languages from Ethiopian. The reason for not testing a related proposed parsing prototype of another language is due to the focus of this research is developing sentence-parsing prototype to Hadiyya language. The other limitation for the proposed and developed SASPFHL is the system not computing semantic parsing for Hadiyya language. This research is conducted for the purpose of academic study, thus it cannot directly applicable for integration with commercial tools. Therefore, SASPFHL can be input in multilingual text processing software in the feature.

1.6. Significance of the Study

This study gives the computer system designer to bind the Hadiyya Language as one of computational language implementation area for multi-lingual application development.

The Hadiyya language speakers (users) will benefit from this study by getting the chance of having their own language parser.

This study may assist other researchers by giving a basic idea on how to process automatically spoken language constituents i.e. word, phrase and sentence, according to grammatical structure of a given language and to farther conducting extended language analysis like syntactic word discourse ambiguity, semantic analysis to Hadiyya language.

The syntactic processing can be implemented for many applications of syntax integrated real application. For example, spelling checkers, grammar analysis, question and answering system, teaching natural language processing, automatic dialogue act recognition, auto-sentence completion systems and other areas of computational linguistics. Therefore, it is an input for feature research.

The developed prototype supposed to guide with syntactic parsing of the Hadiyya Language to support learning and processing grammatically structured text for native or non-native speakers when integrated to text processing systems.

1.8. Organization of the Thesis

This study incorporates six chapters. The first chapter of the study includes introduction, statements of the problem, objectives and methodology. The second chapter elaborate about the existing related literature to the study. Parsing technology selected to develop a parser. This chapter gives in detail explanation about automatic parsing of natural language (a sentence) and concepts of supervised machine learning technique the annotated or labeled corpora.

The third chapter, Hadiyya language grammar or syntactic structure of a sentence and parts of speech included to the words those constituents' phrases for further understanding in developing a parser knowledge base. The forth chapter is about parser developing strategies. Chapter five gives experiment result and discussion. The evaluation of the developed parsing system prototype is described with experimental data set or the test set.

Finally, chapter six presents conclusion and recommendations of the possible feature work.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

Language is one of the fundamental aspects of human behavior and is a crucial component of communication media in our lives. Natural Language Understanding system is computer system that can recognize textual and dialog based human communication language with pre-coded knowledge base on the grammatical structure of the language text.

To be an understanding system, the speech recognizer would need to feed its input to a natural language understanding system, producing what is often called a spoken language understanding system (Allen, 1995).

Parsing is the process of analyzing the grammatical structure of formal linear representation of a language. According to (Dick Grune, 2008), linear representation can be natural language sentences, computer program, a sequence of geological strata, a piece of music, actions in ritual behavior, in short any linear sequence in which the preceding elements in some way restrict the next element.

Parsing can be defined as the decomposition of complex structures into their constituent parts, and parsing technology as the methods, the tools and the software to parse automatically (Harry Bunt, 2005). The complex structures under Automatic Sentence Parsing is a sentence in natural language. ‘Parsing’ is the term used to describe the process of automatically building syntactic analyses of a sentence in terms of a given grammar and lexicon. The resulting syntactic analyses may be used as input to a process of semantic interpretation, (or perhaps phonological interpretation, where aspects of this, like prosody, are sensitive to syntactic structure) (Pulman, 1991).

The parse structure that lies in a grammar plays a great role as a module for the analysis of a language semantics, pragmatic and discourse processing, which constitutes in building the real applications such as text summarization, question answering and machine translation. Thus, parsing is crucial in natural language processing systems and the accuracy of the parses have much impact on implementation of an application as a whole (Alexander Clark, 2010).

A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb (Stanford, 2016).

According to (Tomita, 1991), the parsing technology is a program that synthesize tree like structure from an input sentences. There are several reasons for working on study on automatic natural language processing. One reason, is as stated in the book (Alexander Clark, 2010) that, from the earliest days of machine translation in the 1950s attempts to build parsers have adopted a 'procedural view of language'. In fact, that developing syntax guiding system is not simply a consequence of the machine program, but it is necessary to bind the theoretical works of the linguistics syntax into practical use. Computational parsing is the application of well-founded linguistic theory for natural language processing.

Another reason for building a parser is research on automatic natural language processing gained strength and self-confidence for a numbers of decades, as it has been clearly shown that working system can be built. A context-free phrase structure grammar (CFG) is a formal model for describing whether a given string can be assigned a particular constituent structure. Given a set of syntactic categories, the CFG uses a set of productions to say how a phrase of some category *A* can be analyzed into a sequence of smaller parts $\alpha_1 \ldots \alpha_n$. But a grammar is a static description of a set of strings; it does not tell us what sequence of steps we need to take to build a constituent structure for a string. For this, we need to use a parsing algorithm.

There are several other dimensions, which are useful to characterize the behavior of parsing algorithms. One can characterize their search strategy in terms of the characteristic alternatives of depth first or breadth first. Orthogonally, one can characterize them in terms of the direction in which a structure is built: from the words upwards (‘bottom up’), or from the root node downwards (‘top down’). A third dimension is in terms of the sequential processing of input words: usually this is left-to-right, but right-to-left or ‘middle-out’ strategies are also feasible and may be preferable in some applications (e.g. parsing the output of a speech recognizer) (Pulman, 1991).

2.2. Approaches to Automatic Sentence Parsing

In natural language processing systems machine learning (ML) methods are applied to combat a number of problems to the natural language understanding and its formal representation. The application of ML in natural language processing currently applied in wide range to the speech recognition through morphological analysis and syntactic parsing, to the complex text and discourse understanding. The most common applicable types of ML methods are Supervised and Unsupervised techniques. This are explained as follows:

2.2.1. Supervised Machine Learning Method

Supervised learning entails learning a mapping between a set of input variables X and an output variable Y and applying this mapping to predict the outputs for unseen data. Supervised learning is the most important methodology in machine learning and it also has a central importance in the processing of multimedia data (Alexander Clark, 2010).

The defining characteristic of supervised learning is the availability of annotated training data. The name invokes the idea of a 'supervisor' that instructs the learning system on the labels to associate with training examples. Typically, these labels are class labels in classification problems. Supervised learning algorithms induce models from these training data and these models can be used to classify other unlabeled data. Supervised learning techniques are more powerful than unsupervised techniques because the availability of labelled training data provides clear criteria for model optimization.

2.2.2. Unsupervised Machine Learning Method

Unsupervised work attempts to induce phonological or morphological structure without labeled training data (Daniel Jurafsky, 2009). This unsupervised system could then learn a grammar on its own. This learning technique require large learning pattern from the grammar and thus it is complex for the normal user.

2.3. Strategies for Parsing

Parsing algorithms are usually designed for classes of grammar rather than tailored towards individual grammars. There are several important properties that a parsing algorithm should have if it is to be practically useful. It should be 'sound' with respect to a given grammar and lexicon; that is, it should not be assigned to an input sentence analyses which cannot arise from the grammar in question. It should also be 'complete'; i.e., it should assign to an input sentence all the analyses it can have with respect to the current grammar and lexicon.

Another characteristic of parsing algorithms are the algorithm should also be 'efficient', entailing the minimum of computational work consistent with fulfilling the first two requirements, and 'robust': behaving in a reasonably sensible way when presented with a sentence that it is unable to fully analyze successfully (Pulman, 1991).

The characteristics of parser can be built on the three dimensions as stated on the parsing techniques of the introductory survey by (Pulman, 1991). The first dimension to the algorithms of a parser is based on search strategy in terms of the characteristic alternatives of depth first or

breadth first. The second dimension is orthogonally, one can characterize them in terms of the direction in which a structure is built: from the words upwards ('bottom up'), or from the root node downwards ('top down'). The third dimension is in terms of the sequential processing of input words: usually this is left-to-right, but right-to-left or 'middle-out' strategies are also feasible and may be preferable in some applications (e.g. parsing the output of a speech recognizer).

2.4.1 Top down parsing

The top down algorithm is very simple. It begins with the start symbol, in this case, S, and see what rules it figures in as the root. Then we look at the daughters of these rules and see whether the first one matches the next word in the input. If it does, we do the same for the remaining daughters. If not, then it goes through the same loop again, this time matching the daughters of the rules already found with mothers of other rules.

2.4.2. Bottom up parsing

This type of algorithm behaves by matching words to the right hand side of rules, and then matching the resulting symbols, or sequences of symbols, to the right hand sides of rules until we have an S node symbol covering the whole sentence. Again, we have magically made the right choice at each point.

2.4.3. Chart Parsing

Chart parsing is family of parsing algorithms, all based on the data structure of charts, which is a set of known facts called edges. The algorithm is descended from the Earley algorithms. The parsing process uses inference rules to add new edges to the chart, and the parsing is complete when no further edges can be added. For efficiency most practical parsing algorithms are implemented using a 'well-formed substring table' (wfsst) to record parses of complete sub constituents. This means that duplicate paths through the search space defined by the grammar are avoided. Since there may be many such paths, use of such a table can sometimes mean the difference between processing times of seconds, as opposed to minutes or even hours.

To illustrate, consider the process of parsing a sequence analyzable as

... (1) V (2) NP (3) PP (4)

(e.g. '... saw the man in the park') given rules in the grammar like

VP →V NP

VP → VP PP

NP → NP PP

In a top down parsing regime, there will be two points at which the first of these rules will be used to predict an NP beginning at position 2.

Using a wfsst, already we have recorded the fact that there is a PP from 3 to 4 and instead we simply use that information to build a VP from 1 to 4 of the form [VP [VP PP]]. Thus we have saved ourselves at least two re-computations. If this does not sound very impressive, the reader is invited to work through the steps involved in parsing a sequence like:

V NP P NP P NP P NP

(e.g. 'saw the man in the park with a telescope on Friday') given the above rules, along with P → P NP, and to check how many times the same constituents are reparsed when a wfsst is not being used. For constructions like this, the number scan grows very rapidly indeed. Use of a wfsst here is essential if parsing algorithms are to be implanted in a practically usable form.

A further technique is to keep a record of sub constituents that have NOT been found beginning at a particular point in the input. For example, in the earlier 'saw the man in the park' example, when the VP of the form [VP V [NP [NP PP]]] is found, it will, via the second rule, cause a prediction of a PP beginning at position 4. This prediction fails, because we are at the end of a sentence. When the VP of the form [VP [VP [V NP]] PP] is found, it too will cause the same prediction, via the same rule. While in this case discovering that we have no more PPs is fairly trivial, this will not always be so, and a lot of re-computation can be saved by also checking that, when looking for a constituent C at position P, we have not already tried and failed to find C at P (Dick Grune, 2008).

2.4. Grammatical Formalisms

Many grammatical formalisms can be used by the computational linguistics in modeling natural languages. For example, grammar formalisms include combinatory categorical grammars (CCGs), lexicalized tree-adjoining grammars (LTAGs), link grammars (LG), context sensitive grammars (CSG), Context- Free Grammars (CFG) and Probabilistic Context-Free Grammar (PCFG). To these lexicalized formalisms each word can be thought of as a tree fragment; the full grammatical analysis of a sentence is formed by specifying how and in what order the fragments associated with each word in a sentence combine.

Words may themselves be ambiguous between different parts of speech, here differing tree fragments. In these grammar formalisms parsing a sentence is just deciding on which part of speech to assign to each word. Given this property of these grammar formalisms, perhaps some way can be found to extend the inside/outside algorithm appropriately so that its objective function

maximizes the probabilities of strings of part-of-speech tagged words. If so, it is just a matter of extending the search space to handle the large number of complex part-of-speech structures of lexicalized grammars. The most common and widely used formalisms, Context- Free Grammars and Probabilistic Context-Free Grammar(PCFG) are discussed below as follows: -

2.4.1. Context Sensitive Grammar

A context-sensitive grammar (CSG) is an unrestricted grammar in which every production has the form $\alpha \rightarrow \beta$ with $|\beta| \geq |\alpha|$ (where α and β are strings of nonterminal and terminals). The concept of context-sensitive grammar was introduced by Noam Chomsky in the 1950. In every derivation the length of the string never decreases.

The term" context-sensitive" comes from a normal form for these grammars, where each production is of the form $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$, with $\beta \neq \epsilon$. They permit replacement of variable A by string β only in the" context" α_1 - α_2. Thus, CSG is more restricted to the production rules.

2.4.1. Context-Free Grammars

The most commonly used mathematical system for modeling constituent structure in CFG English and other natural languages is the Context-Free Grammar, or CFG. Context-free grammars are also called Phrase-Structure Grammars, and the formalism is equivalent to what is also called Backus-Naur Form or BNF (Daniel & Martin, 2009).

A context-free grammar consists of a set of rules or productions, each of which expresses the ways that symbols of the language can be grouped and ordered together, and a lexicon of words and symbols. For example, the following productions express NP that a NP (or noun phrase), can be composed of either a ProperNoun or a determiner (Det) followed by a Nominal; a Nominal can be one or more Nouns.

NP → Det Nominal

NP → ProperNoun

Nominal → Noun | Nominal Noun

Context-free rules can be hierarchically embedded, so we can combine the previous rules with others like the following which express facts about the lexicon:

Det → a

Det → the

Noun → flight

The symbols that are used in a CFG are divided into two classes. The symbols that correspond to words in the language ("the", "nightclub") are called terminal symbols; the lexicon is the set of

rules that introduce these terminal symbols. The symbols that express clusters or generalizations of these are called non-terminals. In each context-free rule, the item to the right of the arrow (→) is an ordered list of one or more terminals and non-terminals, while to the left of the arrow is a single non-terminal symbol expressing some cluster or generalization. Notice that in the lexicon, the non-terminal associated with each word is its lexical category, or part-of-speech. The formal language defined by a CFG is the set of strings that are derivable from the designated start symbol. Each grammar must have one designated start symbol, which is often called S. Since context-free grammars are often used to define sentences, S is usually interpreted as the "sentence" node, and the set of strings that are derivable from S is the set of sentences in some simplified version of English.

Let's add to our list of rules a few higher-level rules that expand S, and a couple of others. One will express the fact that a sentence can consist of a noun phrase followed verb phrase by a verb phrase:

S → NP VP I prefer a morning flight

A verb phrase in English consists of a verb followed by assorted other things; for example, one kind of verb phrase consists of a verb followed by a noun phrase:

VP → Verb NP prefer a morning flight

Or the verb phrase may have a verb followed by a noun phrase and a prepositional phrase:

VP → Verb NP PP leave Boston in the morning

Or the verb may be followed by a prepositional phrase alone:

VP → Verb PP leaving on Thursday

A prepositional phrase (PP) generally has a preposition followed by a noun phrase. For example, a very common type of prepositional phrase in the ATIS corpus is used to indicate location or direction:

PP → Preposition NP from Los Angeles

The formal description of a context-free grammar and the language it generates discussed as follows. A context-free grammar G is defined by four parameters N, Σ, P, S:

N a set of non-terminal symbols (or variables)

Σ a set of terminal symbols (disjoint from N)

R a set of rules or productions, each of the form $A \rightarrow \beta$, where A is a non-terminal, β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$

S a designated start symbol

A language is defined via the concept of derivation. One string derives another one if it can be rewritten as the second one via some series of rule applications.

if $A \rightarrow \beta$ is a production of P and α and γ are any strings in the set $(\Sigma \cup N)*$, then we say that $\alpha A\gamma$ directly derives $\alpha \beta \gamma$, or $\alpha A\gamma \Rightarrow \alpha \beta \gamma$.

Derivation is then a generalization of direct derivation:

Let $\alpha 1, \alpha 2, \ldots, \alpha m$ be strings in $(\Sigma \cup N)*$, $m \geq 1$, such that $\alpha 1 \Rightarrow \alpha 2, \alpha 2 \Rightarrow \alpha 3, \ldots, \alpha m-1 \Rightarrow \alpha m$

We say that $\alpha 1$ derives αm , or $\alpha 1 \Rightarrow* \alpha m$.

We can then formally define the language L_G generated by a grammar G as the set of strings composed of terminal symbols which can be derived from the designated start symbol S.

$L_G = \{w|w \text{ is in } \Sigma * \text{ and } S \Rightarrow* w\}$

The problem of mapping from a string of words to its parse tree is called parsing.

2.4.2. Probabilistic Context-Free Grammars (PCFG)

The Probabilistic Context-Free Grammars is a context free grammar that associates a probability with each of its productions head rules. It generates the same set of parses for a text that the corresponding context free grammar does, and assigns a probability to each parse. The probability of a parse generated by a PCFG is simply the product of the probabilities of the productions used to generate it. A probabilistic context-free grammar (PCFG) is a context-free grammar in which every rule is annotated with the probability of choosing that rule. Each PCFG rule is treated as if it were conditionally independent; thus the probability of a sentence is computed by multiplying the probabilities of each rule in the parse of the sentence (Daniel & Martin, 2009).

This grammar formalism models uses linguistic knowledge to condition the probabilities of context-free grammars. These PCFG models, which in essence augment CFG grammar rules with probabilistic applicability constraints. It is based on the hypothesis that the inability of CFGs to parse with giving statistical data information to the parsing production algorithms to acquire accuracy of CFGs to model crucial aspects of linguistic structure relevant to the appropriate selection of the next grammar rule at each point within a context-free derivation. Probabilities in the standard context-free model are conditioned only on the type of non-terminal that the grammar is about to expand. The key idea of these models are that grammars required for parsing provided sufficient linguistic context to adequately model the probabilities of rule expansion.

2.5. Review on related works

With the advancement in technology to understand natural language, a number of researchers contribute for the study of different nations of the world. To process the natural language with computer technology the researchers developed different language parsers using different methods and constructing different algorithms. In this section of study, the literatures that has related to the proposed study will be presented, i.e. the studies, which uses supervised techniques with rule, based approaches to develop natural language parsing. This review will not explore about unsupervised machine learning technique and semantic analysis of natural language. The following study is align with the proposed study and its content is evaluated based on the type of algorithm used, the parcentage of performace evaluation bing attained and the type of technology used for implementation.

(Goldberg, 2011) was developed a parser Automatic Syntactic Processing of Modern Hebrew. As the researcher, the systems cnstructed is capable of parsing naturally occurring Modern Hebrew text. The researcher builds Modern Hebrew dependency annotation guideline and a dependency Treebank. The algorithm used for expermant is a greedy, bottom up dependency parser, based on the principle that easy decisions should be made before more difficult ones. The performance accurecy of parsing result obtaind with accuracies of 77% F1 (constituency) and 79.5% UAS (dependency) when starting from raw text, and 85% F1, 85% UAS when assuming the correct word-segmentation.

(Babarczy, et al., 2005) proposed Hunpars: A Rule-based Sentence Parser for Hungarian. The researchers used purely rule-based approach for developing a parser(Known as Hunpars). Mainly the parser able to process automatically for Hungarian natural language sentences. As indicated in the study, syntactic parser was developed following morphological tagging and part-of-speech disambiguation. The constitunts for the paresr was: A phrase-structure grammar, Lexical databases and searching algorithms. Performance testing was therefore carried out manually thus more prone to error.

Syntactic parsing of the grammar of natural language different unpublished BA and MA thesis have also been conducted on the language(See (Tadesse, 2015)).For instance a senior essay on Hadiyya Verb Morphology written by Moges (1984), Hadiyya phonology by Haileyesus (1984), Nominalization Pattern in Hadiyya by Desta (1989) and “The Morphology of Hadiyya” by Garkebo(2007). were presented to the department of Linguistics and Philology at Addis Ababa

University. In addition, scholar (Sim, 1989) was studied mathematical operation of unification; namely Head-driven Phrase Structure Grammar (HPSG). notional analysis, he made an attempt to propose several revisions of the formalism and address theoretical aspect of the linguistic contribution.

(Diriba, 2002) has developed an automatic sentence parser for Oromo Language using the technique of Supervised Learning. The researcher tried to develop a simple automatic sentence parser for Oromo language. The study was conducted with application of modified chart algorithm and morphology analyzer is developed for identifying the structure. The designed parser is unknown how the grammar formalism considered. He reported that by using an Intelligent (hybrid of Rule-based and supervised learning) the result obtained was 95% on the training test and 88.5% on the test set, but using the small manually parsed sentences the achievement will not be succeeded with the system built. The method is ad-hoc cross checking against human (linguists) parser. The study was not addressing all type of sentences, i.e. limited to parsing for few simple sentence of Ormo Language.

2.6. Conclusion

It would be convenient, and intellectually tidy, if automatic natural language parsing could be discussed without any extended reference to the history of theoretical linguistics, as if computational analysis had its own autonomous life; and it is true that at times artificial intelligence workers have disregarded the concerns of contemporary theoretical linguistics. For others, however, autonomous parsing must be intellectually disreputable, mere amateur posturing: for them computational parsing ought to be the application of (well-founded) linguistic theory (Jones Karen, 1985).

Many tabular parsing methods are capable of computing and storing exponentially many parses using only polynomial time and space. Tabular parsing, invented in the field of computer science in the period roughly between 1965 and 1975, also became known later in the field of computational linguistics as chart parsing. Tabular parsing is a form of dynamic programming, a standard paradigm in the design of computer algorithms (Alexander Clark, 2010).

The literature studied under this section was showing the attainment of good result with performance evaluation. This result leads or encourage to do the application of supervised rule based techniques for automatic natural understanding system is safe to be built to the desired language of Hadiyya.

Automatic Rule based sentence parsing is pre-dominent supervised method and the most successful natural language processing technique. The wholl achivement through an experment was suceeded greater than 50% with performance evaluation, and thus it encareges this new proposed system to be studed.

CHAPTER THREE

SYNTAX OF HADIYYA LNAGUAGE

3.1. Introduction

The structural rules that bind the sentence in the communication language determins syntax reperentation or formal grammar. In aritifitial language understanding systems for the natural language knowledge representation it is important to present a governing pattern according to pre-existing writing systems in scintific standerdized context. The text of the language is constituents of different sub-parts of sentences. The sentence also contain different parts of speech like word, Noun, Verb, Adverb, Adjective, Prepositions, punctuation marks, numbers that convey semantics for understanding and communication of the language in context.

In this chapter, the basic information on phonetics and other aspects of Hadiyyisa (Hadiyya language) grammar constitunts are discussed. Eventhogh, this study mainly concerned to syntactic structure of Hadiyya language, the aspects that constitutes the sentence of the language need to understand i.e. the orthography and phonology of the word in general. Therefore, the sub-sections in the chapter providing users with some guidance on what the basic orthographic and phonetic rules are in order to write and pronounce words in the Hadiyya language. Furthrmore basic underlying facts about the nature and linguistic building blocks of Hadiyyisa is discussed.

In Hadiyya the writing system is drived from latin based alphabet, but this language comes up with some different representation for the letter as shown in (fig 3.1) below. The reason why Hadiyya languge coded in its own script is due to unique orthographic and phonetic nature of this language required creating a methodology that specifies certain groupings of usages to address variations in meaning and pronunciation. Certain rules had to be developed to group and characterize certain grammatical usages that appear to be peculiar to this language. As can be observed, the descriptions of such grammatical peculiarities below indicate an attempt to do just that, i.e. certain phonetic/grammatical rules. So the fact that there is the difference in pronouncing a word that can have different meanings by simply alternating the level of stress being applied to it could constitute a rule. But the fact that there are plenty of words in a given text that may exhibit such characteristic requires some means to identify and explain the words or syllables involved. One way to do that is to label those peculiar grammatical incidents in the text with something like a symbol. Certain

symbols are randomly selected just for this purpose and each symbol is assigned a rule that is described next to the symbol. In addition to symbols or signs such as the apostrophe (’), double letters are used to indicate specific grammatical/phonetic or transcription rules (Mishago, 2005).

Alphabetical order			Phonemes
	Lower case	Upper case	
1	a	A	/a/
2	b	B	/b/
3	c	C	/tʃ'/
4	d	D	/d/
5	e	E	/e/
6	f	F	/f/
7	g	G	/g/
8	h	H	/h/
9	i	I	/i/
10	j	J	/dʒ/
11	k	K	/k/
12	l	L	/l/
13	m	M	/m/
14	n	N	/n/
15	o	O	/o/
16	q	Q	/k'/
17	r	R	/r/
18	s	S	/s/
19	t	T	/t/
20	u	U	/u/
21	w	W	/w/
22	x	X	/t'/
23	y	Y	/j/
24	z	Z	/z/
25	ch	CH	/tʃ/
26	ph	PH	/P'/
27	sh	SH	/ʃ/
28	'	'	/ʔ/

[source: *(‘Tadesse S. G., 2015)*]

Table 3. 1 Hadiyya Alphabet

3.2. Part of Speech

In parse data structures, there are numerous possible parts of speech that include noun, verb, adj, adv, qual (qualifier), det, prep, subconj (subordinating conjunction), and conj (coordinating conjunction). Some of these are seen in the sample parse tree, where the POS is listed as the first of the nodes’ features.

“The most basic unit of linguistic structure appears to be the word. The word, though, is far from the fundamental element of study in linguistics; it is already the result of a complex set of more primitive parts. The study of morphology concerns the construction of words from more basic components corresponding roughly to meaning units. There are two basic ways that new words are formed, traditionally classified as inflectional forms and derivational forms. Inflectional forms use a root form of a word and typically add a suffix so that the word appears in the appropriate form

given the sentence. Verbs are the best examples of this in English. Each verb has a basic form that then is typically changed depending on the subject and the tense of the sentence" (Allen, 1995).

We can identify four main classes of words in English that contribute to the meaning of sentences. These classes are nouns, adjectives, verbs, and adverbs. Sentences are built out of phrases centered on these four word classes.A word in any of the four open classes may be used to form the basis of a phrase. This word is called the head of the phrase and indicates the type of thing, activity, or quality that the phrase describes.

Traditionally, linguists classify words into different categories based on their uses. Two related areas of evidence are used to divide words into categories. The first area concerns the word's contribution to the meaning of the phrase that contains it, and the second area concerns the actual syntactic structures in which the word may play a role. For example, you might posit the class noun as those words that can be used to identify the basic type of object, concept, or place being discussed, and adjective as those words that further qualify the object,concept, or place. (Allen, 1995).

3.2.1. Noun

The categories of Hadiyya nouns can be inflectional and derivational as stated by (Tadesse, 2015) in the modern Hadiyya documentation and description. Inflection is the grammatical function of words in phrases without altering their meaning. In Hadiyya, nouns are inflected for gender, number, definiteness and case.

Gender

Gender, refers to the sexual distinction between male and female. In Hadiyya, animate nouns distinguish between masculine and feminine gender biologically using totally different lexemes. Thus, most kinship terms and common domestic animate nouns have superlative forms distinguishing masculine and feminine genders as shown in the following table: -

Feminine	English meaning	Masculine	English meaning
[gannítt∫o]NN	'male'	[faráʃʃo]NN	'horse'
[landíttʃo]NN	'girl'	[beéto]NN	'boy'
[Saájja]NN	'cow'	[baára]NN	'ox'

Table 3. 2 Nouns Inflection for gender

The suffixes -e and -o are attached to proper nouns to express the feminine and masculine genders respectively. The following table shows some of proper nouns for illustration.

Feminine Proper Nouns	***Masculine proper nouns***
[dambál-e]NNP	[dambál-o] NNP
[dileéb-e] NNP	[dileéb-o] NNP
[dobaám-e] NNP	[dobaám-o] NNP
[liiráns-e] NNP	[liiráns-o] NNP

Table 3. 3 Proper Nouns in Hadiyya

Number

Nouns show a three-way number distinction among singular, plural and general number values. numerals and quantifiers are added to disambiguate such indistinct translation.

The Singular

The singular form is expressed by the morpheme -ittʃ and its allomorph -itʃ. examples of singular nouns formed by the use of suffix -ittʃ.

General number	Singular	English meaning
[góta]NN	[got-íttʃ]NN	‘a/the hyena’
[hálla]NN	[hall-íttʃ]NN	‘a/the donkey’

The plural

The majority of plural nouns are formed by attaching the suffix -uww to the citation form. The final vowel of the citation form is replaced by the plural suffix.

General number	Plural	English meaning
[daánna]NN	[daann-uww-]NNS	‘judges’
[gitánna]NN	[gitann-uww-] NNS	‘heros’
[k’áwwa]NN	[k’aww-uww-] NNS	‘foolishes’
[ʔaájja]NN	[ʔaajj-uww-] NNS	‘sisters’

Uncountable nouns have only general form. Such nouns are neither inflected for plural nor for singular suffix. However, a large amount is expressed by using ʔarạk’a ‘much.

General number	uncountable nouns + ʔarak’a [‘much’]	English meaning
[barạda]NN	[ʔarak’-ị-barạda]NN	‘much ice’
[wóʔo]NN	[ʔarak’-ị-wóʔo]NN	‘much water’

Definiteness

Definiteness refers to the grammatical category used to demonstrate whether a noun is known or unknown. The indefinite reference is not morphologically marked in Hadiyya. Definiteness (DEF) is marked morphologically by -oom.

Indefinite Definite	English meaning	Definite	English meaning

[beéto]NN	'boy'	[beet-oom]NN	'boy-DEF'
[fellá?a]NN	'goats'	[fella?-oom]NN	'goats-DEF'
[minnúwwa]NN	'houses'	[minn-uww-oom]NN	houses-PL-DEF

Case

In Hadiyya the 'peripheral' cases include dative, ablative, locative, instrumental, and commutative.

The absolutive case

The absolutive case is the grammatically unmarked form of the noun. below are sample lists of masculine and feminine nouns in absolutive case. Nouns occurring in the absolutive case too, end in one of these vowels in -a,-o or -e but never either in i or u.

Masculine nouns	Feminine nouns
[baárạ]NN	[saájjạ]NN
'ox'	'cow'

The Nominative Case

The nominative is used to encode the subject of both transitive and intransitive sentences. The nominative of masculine nouns ends in an extremely short and devoiced -i.

Example A. mantʃ-ị kaʃári gereé-ttʃọ dur-ú-kk-o

man.SG-NOM red sheep-SG.ABS slaughtwer.3SG.M-PV-ASM.TV

'A/the man slaughtered a red sheep.'

B. [lánd-ị]NN [waar-am-ú-kk-jjo]VBDN

'The girls didn't come'

Nominal derivation and compounding

Derivation is the morphological process by which new words are formed from other words or roots. Hadiyya has also a set of derived nominals. Derived nominals are formed by derivation and compounding.

Nominal derivation

Nouns are formed from roots mainly by the use of suffixes. Below are some examples of abstract nouns derived from verbal/adjectival bases via suffixation of -ooma,-at, -an, -imm, -o, -a, -ttʃ and -eena.

Lexical base	Derived nominals	English meaning
bat'-	[bát'-o]NN	'job'
wotʃ-	[wótʃ-a]NN	'conflict'
mar-	mar-eéna	'to go'

k'ananaaʔ-	k'ananaaʔ-imm	'reading'

Compound nouns

The process of compounding in which new words are formed by combining two or more independent lexical items. Nominal compounds in Hadiyya consist of two substructures, the modifier is either an adjective or a verb and the head is a noun.

Examples of the Noun + Noun combinations:

N + N	Nominal compounds	English meaning
horoóre 'head' + ʔódda 'hair'	horooʔlódda	'hair'
láro 'cows' + ʔaágga 'entering'	laʔliʔaágga	'night'

Adjective + Noun combinations

Adj + N	Nominal Compounds	English meaning
ʔer- 'good' + súmmo 'name'	ʔersúmmo	'well-known, famed'
lob- 'great' + mánna 'men'	lommánna	'old people'
Exocentric compounds	Nominal compounds	English meaning
máto 'one' + maára 'meat'	matmaára	'close relative'
t'aára 'metal' + gána 'hitting'	t'aargána	'engagement
ʔánga 'hand' + hoffiítt'a 'little'	ʔangahoffiítt'a	'poor'

Many proper nouns are compounds. Such compounds are formed by using derived nominals as a second component. Here is an examples:

N + N = N (Proper nouns)	Nominal compounds	English meaning
leho 'death + hoóro 'protect'	lehoóro	'savior, rescuer'
mánna 'men' + ʔiítto 'love'	manniítto	'popular, enthusiast'

Numeral + Noun combinations

Numerals + N	Nominal Compounds	English meaning
lámo 'two' + foóre 'soul/life'	lamfoóre	'pregnant'
lámo 'two' + hákk'a 'wood'	lamákk'a	'double'

Possessive Pronouns + Noun combinations

Possessive Pronouns + N	Nominal Compounds	English meaning
kị- 'your' + ʔaájja 'sister'	kijjaájja	'your sister'
ʔi- 'my' + ʔaájja 'sister'	ʔijjaájja	'my sister'

3.2.2. Pronoun

According to the researcher (Tadesse, 2015), the categories of pronouns in Hadiyya are possessive, demonstrative, reflexive, interrogative, restrictive and vocative.

Personal pronouns

Gender distinction in personal pronouns is made only in the third person singular forms. honorific pronouns kiʔne (2SG) & ʔisse (3SG) are used when speaking about respected and old people.

Consider the following examples with the subject pronoun:

A) [ʔáni̱]PRP [los-aán-tʃo]NN

'I am a student.'

B) [ʔáni̱] PRP [ʔit-ú-mm-o]VBD

'I ate.'

The object pronouns

Like nouns, the pronouns devoice their terminal vowels in the absolutive case. Here is an example.

A) [wiʃ-íttʃ-i̱]NN [ʔítt'o̱]PRP [holl-ú-kk-o]VBD

The dog chased him.

B) [ʔáni̱]PRP [keésę̱]PRP [laʔ-oo-mm-ó-jjo]VBD

'I do not know you.'

The dative pronouns

The dative pronouns in Hadiyya are formed by the suffix -ina and its allomorph -na. The following are the Dative pronouns in Hadiyya. Example:

A) daánn-i̱ ʔitt'é-na lab-ú-kki̱ ʔogáto̱ ʔuww-oó-kk-o

'The judge has given the judgment on his own way'

B) kába gosáʔni bikk-ína ʔaaloʔ-ísa ʔi-ína kúr-e

now abduction.GEN about-DAT good-SIM 1SG-DAT tell.2SG-IMP

'Now, explain to me about gosano (abduction) in a detailed manner.'

The possessive pronouns

Genitive constructions are formed by using independent forms of genitive pronouns. The head noun involved in the phrase is lókko 'leg'.

Examples: ʔi lókko 'my leg'

ki lókko 'your (SG) leg'

ʔítt'i lókko 'his leg'

ʔísi lókko 'her leg'

The ablative pronouns

Like nouns, ablative pronouns are formed by the suffix -iinse.

Example: A) ku ʔassé-ttʃ-i̱ ʔitt'uww-iínse ʔasseʔ-am-ú-kk-o

'This letter was sent from them.'

B) ʔítt'i̱ niinn-iínse diinátę̱ lik'aájją̱ ʔaaʔ-ú-kk-o

'He borrowed money from us.'

To depict comparative expressions structures the morpheme -iinse is suffixed to pronouns.

Example: A) ʔáti̱ ʔiin-iínse k'oóra

2SG.NOM 1SG-ABL clever.COP

B) ʔísi̱ kiinn-iínse danaamó-tte

3SG.F.NOM 2SG-ABL beautiful-COP

The reflexive pronouns

The reflexive pronouns are formed by suffixing the word gága 'self' to the genitive pronoun forms to render the meaning 'one self' and horoóre 'head', The reflexive pronoun shows neutralization of gender and number features. Illustration example:

A) ʔítt'i ʔitt'i gágą̱ t'agaʔl-ú-kk-o

'He hung himself.'

The vocative

The vocatives can be used to call a person over a distance or, to address a friend. vocatives are restricted only to 2SG.

Examples: (a) [kaá]PRP 'You! (SG.M)'

(b) [taá]PRP 'You! (SG.F)'

3.2.3. Verb

The verbal root, which takes inflectional and derivational affixes in the verb conjugation, is a bound morpheme too. The symbols C stands for Consonant and V stands for vowel. The verbal roots can be categorized as monosyllabic, disyllabic or trisyllabic.

a) monosyllabics (CVC-, CVCC-, CVVCC-)

CVC- ʔed- 'add'

CVCC- holl- 'chase'

CVVCC- dooʔl- 'choose'

b) Disyllabics (CVCVCC-, CVCVVC-, CVCVVCC-, CVCCVC-, CVCCVVC-, CVVCCVC-, CVVCCVVC-)

CVCVCC- galatt'- 'thank'

CVCVVC- firiit'- 'gallop'

CVCVVCC- ʔuʃeett'- 'rest'

CVVCCVC- taakkeʔ- 'walk'

(c) Trisyllabic (CVCVVCVC-, CVCCVCVVCC-, CVCVVCCVC-, CVCVVCCVC-, CVCVCVVC-, CVCVCVC)

CVCVCVC- hit'it'aʔ- 'pull'

CVCVVCVC- ʔadʒaaban- 'become angry'

CVCCVCVC- ʔankakar- 'crawl'

Verb inflection

The expression of inflectional categories in verbs is exclusively suffixal. Verbs can be marked with suffixes for three types of temporal aspects, the imperfect, the simple perfect, and the perfect, together with suffixes for the person/number of the subject.

Subject agreement marking

Consider now the occurrence of these subject agreement markers in the following perfective paradigms. Either of the following verbal roots is used as an example in this section: ʔuww- 'give', ʔaanʃ- 'wash', mass- 'take', dooʔl- 'select', diss- 'put', badd- 'be afraid', lik'itʃ'- 'swallow', suunk'- 'kiss', ʔuundʒ- 'leave', ʔuunt'- 'beg'.

Consider the distribution of these suffixes in the following examples:

(a) goógọ bat'-am-ú-kkị mann-ína diinátẹ mik'-am-u-kk-ó-jjo

road.ABS work-3PL-PV-REL.ASM man-DAT money.ABS pay-3PL-PV-ASM-TV-NEG

'They did not pay the money for them who built the road.'

(b) daanéttʃọ bagad-í-nne ʃi-n-ú-mmị mánn-ị neesé-tte

elephant.SG.ABS spear-EP-INST kill-1PL-PV-ASM.REL man-NOM 1PL-COP

'We are the guys who killed an elephant with a spear.'

Aspect marking

"different ways of viewing the internal temporal constituency of a situation" and further states that "[a]spect is not concerned with relating the time of the situation to any other time-point, but rather with the internal temporal constituency of the one situation".

Perfective

Perfective aspect is indicated by the vowel -u- following the subject agreement marker and preceeding the additional subject marker consonants as the following paradigm illustrates for the verbs diss- 'put', lik'itʃ'- 'swallow', suunk'- 'kiss' and ʔuunt'- 'beg'.

Perfective marker

Person	1SG	1PL	2SG	2PL	3SG.M	3SG.F	3PL	3HON
Perfective	-u	-u	Ø	-o	-u	-o	-u	-o

Aspect vowels

Example: (a) kabálla ʔarákį goógǫ taakkeʔ-n-ú-mm-o

today much road.ABS walk-1PL-PV-ASM-TV

'We traveled a very long way today.'

(b) ánį beebálla kojj-ína búnǫ kaas-ú-mm-o

1SG.NOM yesterday guest-DAT coffee.ABS put.on.fire.1SG-PV-ASM-TV

'I made coffee for the guests yesterday.'

The Present perfect aspect

The present perfect aspect, reveals a relation between two time-points: the time of the state resulting from a preceding incident/event and the time of that preceding event. The perfect aspect is marked by the vowel -aa and its allomorph –oo.

Affirmative and negative paradigms between perfective and perfect

Aspect	Affirmative	Negative
Perfective (a)	dooʔl-ú-mm-o choose.1SG-PV-ASM-TV	dooʔl-u-mm-ó-jjo
Perfect (b)	dooʔl-aá-mm-o choose.1SG-PV-ASM-TV-NEG	choose.1SG-PF-ASM-TV
Perfective (c)	ʔuunt'-ú-kk-o beg.3SG.M-PV-ASM-TV	ʔuunt'-u-kk-ó-jjo
Perfect (d)	ʔuunt'-aá-kk-o beg.3SG.M-PV-ASM-TV-NEG	beg.3SG.M-PF-ASM-TV

Imperfective aspect

The vowel oo, which marks perfective in most HEC languages, is used to mark imperfective aspect in Hadiyya. when time adverbs such as, soódo 'tomorrow', maaróʔo 'tonight', matmatiʔammáne 'sometimes', hundiʔammáne 'always', daridára 'every morning' are added, imperfectives depict only one reading, either future or habitual actions respectively.

(a) maaróʔo mar-oó-mm-o

night go.1SG-IPV-ASM-TV

'I will go in the evening.'

(b) matmáti ʔammáne gaʔn-oó-kk-o

sometimes rain.3SG.M-IPV-ASM-TV

Progressive aspect

In Hadiyya, the progressive imperfective aspect which refers to an action running over an extended time is encoded by the suffix -ulla and its allomorph -lla. Note that the all morph -lla is attached to verbs of 2SG and 3SG.M, whereas -ulla is elsewhere. The use of -ulla/lla as progressive imperfective aspect marker, is exemplified in here with the verb waattʃ- 'to swim'.

(a) waattʃ-oo-mm-úlla 'I am swimming.'

(b) waattʃ-i-n-oo-mm-úlla 'We are swimming.'

Tense

"Tense is the grammatical expression of the relation of the time of an event to some reference point in time, usually the moment the clause is uttered." Of the commontense systems "past, present, and future". Hadiyya has only the past. The past tense is formed as complex constructions made up of the main verb followed by the verb heeʔ- 'be' in the Perfective aspect, used as an auxiliary.

Past paradigms

	diss- 'put'	guull- 'finish'	waar- 'come'
1SG	dissaaheeʔúmmo	guullaaheeʔúmmo	waaraaheeʔúmmo
1PL	dissinaaheeʔnúmmo	guullinaaheeʔnúmmo	waarnaaheeʔnúmmo
2SG	dissitaaheeʔlítto	guullitaaheeʔlítto	waattaaheeʔlítto

3.2.3. Adjectives

In Hadiyya adjectives constitute a separate word class. Adjectives are characterized by having one of the suffixal elements -a, -o, and –e. Adjectives can occur following the degree word lobakáta 'very/too much' given as example.

(a) [lobakáti̱]RB [ʔamatt'-aam-ína kóbi̱ [dúta]JJ

'A wealthy [person] has got lots of friends.'

(c) [ʔísi̱]PRP [lobakáta]JJ [k'oorá-tte]RB

'She is very clever.'

Below are given examples of adjectives functioning as modifiers.

(a) [gúndi̱]JJ [ʔánn-i̱]NN [ʔoos-ína]RB [darábi̱]NN [mán-tʃo̱]NN [lab-oó-kk-o]VB

'A short father seems of the same age [a friend] for his own sons.'

(b) [k'oók'i]JJ [ʔíll-i̱]NN [barbaróʔo̱]NN [badd-oó-jjo]JJ

'A blind eye is not afraid of a chili powder.'

Derivation

Drived adjectives in the language are derived from nominal and verbal roots. Suffixes added to verbal and nominal roots to form adjectives are -aam, -aalla, -a, -o, and -e. Here is an examples. The suffix -aam is also used to derive a large number of proprietive adjectives, which are used in proverbs. Consider the use of such derived adjectives in the following Proverbs:

(a))ʔaboʔ-aám-i̱ ʔamatt'-aam-ína bat'-oó-kk-o

'A white haired old man kneels down for a wealthy one.'

(b)) ʔallab-aám-i̱ bagad-aámo̱ holl-ú-kk-o

'A sharp tongued chases away the warrior.'

3.2.4. Numerals

Numerals are found in modifying function preceding the head noun. The numerals used in ordinary counting are spesified here :

Cardinals and ordinal numbers from one to ten.

A. Cardinals	B. Ordinals
[Máto]CD 'one'	lútt'i̱ 'first'
[lámo]CD 'two'	láʔmi̱ 'second'
[sáso]CD 'three'	sátt'i̱ 'third'
[soóro]CD 'four'	soóʔli̱ 'fourth'
[ʔónto]CD 'five'	ʔóntɨ̱ 'fifth'
[lóho]CD 'six'	lóhɨ̱ 'sixth'
[lamára]CD 'seven'	lamáʔli̱ 'seventh'
[sadeénto]CD 'eight'	sadeéntɨ̱ 'eighth'
[hónso]CD 'nine'	hónsɨ̱ 'ninth'
[tómmo]CD 'ten'	tómmɨ̱ 'tenth'

Table 3. 4 Numbers in Hadiyya

The use of máto as indefinite pronoun is illustrated below in the following example:

a) gaási doollé-nne mátɨ̱ bat'-aán-tʃ-i̱ tʃ'aná-nne t'aafé?e̱

previous.GEN epoch-TDC one work-AGN-SG-NOM leather.pouch -LOC teff.ABS

ʔijj-aá meéra mar-u-kk-úlla doóʔmi woró-nne máti

carry.3SG.M-CNV market go.3SG.M-PV-ASM-PRG forest.GEN in-LOC one

kabee-ttʃ-í-nne ʔed-amam-ú-kk-o

leopard-SG-EP- COM meet.3PL-RCP-PV-ASM-TV

‘Once upon a time, a farmer who carried a sack of teff (grain) and traveling to the market has come across a leopard.’

Quantifiers

Hadiyya has the following quantifiers:

Quantifires	English meaning
dúta	‘several, many’
hóffi̱	‘little, few’
húnda	‘all’
lobakáta	‘much’
matimáti̱	‘some’
t’aléʔe	‘only’
ʔarák’a	‘much’

3.2.5. Conjunction

Hadiyya has the following phrasal and clausal conjunctive and disjunctive devices on the basis of their functions: coordinative, inclusive, alternative, resultative, adversative (reversing).

Coordinative

Nominal coordination is signaled by lengthening the final vowel of all the constituents which are coordinated. Indicating conjunction through final vowel lengthening is productive device for all the major word classes i.e., nouns, pronouns, verbs and adjectives can be conjoined by such coordinating strategy. These are shown in the following examples respectively.

(a) [ʔítt’i̱]PRP [ka]DT [beet-ína]CCP [kitaábą̱]NN [ʔuww-ú-kk-o]VBD

‘He gave books to this boy.’

(b) ʔítt’i̱ ka beet-ina-á ta land-íttʃo-na-á kitaábą̱ ʔuww-ú-kk-o

‘He gave books to this boy and to this girl.’

Inclusive

A lexical element ʔodíme and a suffixal morpheme -m are used as inclusive conjunctions. illustrative examples:

Phrasal and clauses coordination by –m

(a) mán-tʃ-i̱ ʔítt’i ʔanná-mi̱ ʃ-ú-kk-o

‘The man kiiled his father too.’

(b)kiʔnuwwí-mi̱ danaam-ísa batt’-éhe

‘You too, do it in a good manner.’

Phrasal and clausal coordination by ʔodime

(a) k'amáttʃ-i̱ ʔodíme las-aán-tʃi̱ ʔoogátọ ʔuww-ú-kk-o

'Ape in its part gave the final judgment.'

(b) ʔáni̱ lar-ína hít'ẹ ʔuww-ú-mm-o ʔi meent-íttʃo ʔodíme búna̱ kaass-ó-ʔ-o

'I gave the cows grass, my wife however made coffee.'

3.3. Word order

All the words in Hadiyya terminate in a vowel. Like other related cushitic languages, the basic word order in Hadiyya sentences is SOV in transitive and SV in intransitive sentences. Here are the examples, to demonstrate:

(a) [land-i̱]NNS [ka]DT [ʔeeb-akk'-ám-tʃa̱]NN [sab-am-oó-kk-o]VB

'Girls refuse such type of marriage.'

(b) [ʔáni̱]PRP [gereé-ttʃọ]NN [dur-ú-mm-o]VBD

'I slaughtered a sheep.'

c) [ʔi]PRP$ [beʃ-íttʃ-i̱]NN [waar-oó-kk-o]MDVB

'My friend will come.'

d) [ʔítt'i̱]PRP [geer-oó-kk-o]MDVB

He will run.

3.3.1. Noun Phrase

Noun phrases (NPs) are used to refer to things: objects, places, concepts, events, qualities, and so on. The simplest NP consists of a single pronoun: he, she, they, you, me, it, I, and so on. Pronouns can refer to physical objects, as in the sentence (Allen, 1995). According to the researcher (Sim, 1989) "S-O-V order is the base for Hadiyya."

According to, (Tadesse, 2015) the basic word order within the noun phrase is: Modifier Head. Thus, adnominal modifiers (adjectives, numerals, qunatifires, demonstratives, possessives and the relative clauses), come before the head noun they modify, as illustrated respectively in examples below and the tags JJ used for adjective and NN for Noun representation here in the examples.

(A) Adjective-Head noun

(a) [gúndi̱]JJ [mán-tʃọ] NN 'a short man'

(b) [haréttʃi̱]JJ [míne̱]NN 'new house'

(c) [k'oóri̱]JJ [beétọ]NN 'wise boy'

(B) Numerals-Head noun

(a) [lámị ʔ]JJ [abbaajj-úwwạ]NNS ‘two brothers‘

(b) [lamárị]JJ [saántạ]NN ‘seven weeks‘

(c) [mátị]JJ [mán-tʃọ]NN ‘a man‘

(C) Qunatifier-Head noun

(a) [lobakátị]JJ [ʔoósọ]NN ‘many children‘

(b) [ʔarák’ị]JJ [búttʃạ]NN ‘much soil‘

(c) [dútị]JJ [mánnạ]NN ‘many people‘

(D) Demonstrative-Head noun

(a) [ku=meént-ị] JJ ‘these women’

(b) [tu=land-íttʃọ]JJ ‘this girl‘

(E) Possessive-Head noun

(a) [bak’úttʃi]JJ [lókkọ]NN ‘mules leg‘

(b) [ki]JJ [beʃ-íttʃọ]NN ‘your friend‘

(c) [miʃaámi]JJ [meent-íttʃo]NN ‘Mishaamo’s wife‘

(F) The relative clause-Head noun

(a) [foóręˌ] [ʃ-ú-kkị] [mán-tʃ-ị] [heemáttʃị]JJ [gereé-ttʃọ]NN [ʔeeb-oó-kk-o]

‘First, the person who killed will deliver a black sheep.’

(b) biʔis-ú-kkị mán-tʃ-ị guzumóʔọ marábọ mik’-oó-kk-o

‘The person who involved in murder wiil deliver honey as penality.’

most of the quantifiers occur before the head noun, these two quantifiers occur following the noun they modify as the following examples demonstrate.

(G) ku mánn-ị hund-í-mị ‘All of the people’

(H) wotʃ’-am-ú-kkị mánn-ị t’aléʔe ‘Only the people who are involved in the conflict’

In all of these examples, no other order within the NP is possible with regard to the demonstrtives. As shown in examples which k’eeráʔlị ‘tall’, danaámị ‘good’ and k’adaállị ‘white’ occur between the respective head nouns and the genitive noun/personal pronuons.

(a) [ʔi]PRP [beét-ị]NN ‘my son’

(b) [ki]PRP [meent-íttʃo]NN ‘your wife’

(c) [ʔítt’i]PRP [baár-ị]NN ‘his ox’

3.3.2. Verb

The most frequent word order in intransitive and transitive sentences is respectively, SV and SOV. VS order for intransitive sentences is not attested; however OSV order in transitive sentences is attested. The SOV word order exemplified in (A) can also be uttered in OSV order as in (B).

(A) [ka]DT [ʔeeb-akk'-ám-tʃa̰]NN [lánd-ḭ]NNS [sab-am-oó-kk-o]VBD
'Girls refuse such type of marriage.'
(B) [gereé-ttʃo̰]NN [ʔánḭ]PRP [dur-ú-mm-o]VBD
'I slaughtered a sheep.'

OSV word order is used only when the subject is in focus. As the following examples demonstrate, the preverbal position is reserved for emphasized information.

(C) ʔ ísḭ tʃ'iil-íttʃ-ína wóʔo̰ ʔuww-i-t-ó-ʔ-o ʔádo̰ ʔuww-i-t-o-ʔ-ó-jjo
'She gave water to the child.' (She didn't give the child milk.)
(D) ʔísḭ wóʔo̰ tʃ'iil-íttʃ-ína ʔuww-i-t-ó-ʔ-o ʔádo̰ ʔuww-i-t-o-ʔ-ó-jjo
'She gave water to the child.' (She didn't give to me.)

In command sentences involving the dative, the Subject-Dative complement-the Object-Verb order is most frequently used:

(a) ʔátḭ ʔisé-na diinátę̰ ʔúww-e 'You! Give her money.'

Subjects often occupy the first position of a sentence. If they are moved to pre-verbal position, they are focused.Consider the following examples:

(a) ʔ isé-na hooʃʃóʔo̰ meent-íttʃo ʔuww-i-t-ó-ʔ-o mán-tʃ-ḭ ʔuww-u-kk-ó-jjo
'The woman gave her lunch. The man didn't.' (It is no one but the woman who gave her lunch.)
(b) kojjí-na búna̰ ʔánḭ kaas-ú-mm-o
'I made coffee for the guests.' (It is no one but I made coffee for the guests.)

Temporal nouns often occupy the post subject position of a sentence. If they are positioned in reverbal, they are focused. Compare the focused elements in (a-b), i.e. the subject is focused in (a), whereas temporal noun in (b).

(a) [ʔánḭ]PRP [beebálla]NN [waar-ú-mm-o]VBD 'I came yesterday.'
(b) [beebálla]NN [ʔánḭ]PRP [waar-ú-mm-o]VBD 'I came yesterday.'

The subject of a non-verbal sentence occurs preceding the predicate. For instance, ʔimeentittʃótte ‘my wife’ (a), ʔise-tte ‘it is her’ (b), are predicates of their respective clauses.

(a) ʔoo tte ʔi meent-íttʃó-tte ‘That is my wife.’

(b) ʔi land-íttʃo ʔisé-tte ‘My daughter is she.’

With regard to questions the same word order is attested. The interrogative pronoun occurs in the position where the corresponding the noun/phrase which is questioned. Consider the following questions in (A) with corresponding answers in (B):

A) Questions

(a) [ʔájj-ị]WDT [keésẹ]PRP [haraʔm-ú-kk-o]VBD ?

‘Who helped you?’

(b) [lánd-ị]NNS [hínka]WDT [ʔeeb-akk’-ám-tʃạ]NN [doʔl-am-oó-kk-o]VB

‘Which marriage type do girls choose?’

(B) Possible answers/declarative sentences

(a) [ʔi]POS [beʃ-íttʃ-ị]NN [ʔeésẹ]PRP [haraʔm-ú-kk-o]VBD

‘My friend helped me.’

(b) [lánd-ị]NNS heer-ám-tʃạ [doʔl-am-oó-kk-o]VB ‘Girls choose heerantʃa.’

Dependent clauses usually occur preceding the main clause, as illustrated in the examples below:

(a) [beét-ị]POS [waar-u-kk-aá-re]VB [ʔánị]PRP [mar-oó-mm-o]VBG

‘If the boy comes, I will go.’

(b) [wiʃ-íttʃ-ị]NNP [moóttʃạ]JJ [siid-oó]VB [ʔammáne]RB [lobakáta]RB [muun-oó-kk-o]VB

‘When a dog sees a wild aimal, it barks uninterruptedly.’

3.4. Sentence

On the basis of clause types, sentences are divided into simple and complex sentences. Simple sentences consist of a single main clause while complex sentences comprise one or more subordinate clauses, in addition to a main clause.

According to (Tadesse, 2015), in Hadiyya the following structural categories or sentence types are distinguished: affirmative/negative declarative clauses, interrogatives and imperatives.

3.4.1. Simple Declarative Sentences

A declarative refers to verb forms or sentence/clause types typically used in the expression of statements. In Hadiyya, there is no distinct morphological marker for declaratives. A declarative verb can be affirmative or negative. Each of these declarative sentence types will be dealt in turn.

3.4.1.1. The Affirmative Declarative

The affirmative is expressed by a zero morpheme that contrasts paradigmatically with the negative marker -jjo. The different types of declarative can be formed, such as eventive, stative, informative, active, and passive sentences. Look at the fllowing exampes

(a) [ʃaameéb-i]NNP [las-aán-ʧo]NN
'Shaamebo is a student.'

(b) [ʃaameéb-i]NNP [dʒabb-aán-ʧi]JJ [ʔih-ú-kk-o]VBN
'Shaameebo became sick.'

3.4.1.2. The negative declarative

Hadiyya is among the languages that exhibit different negative markers. All types of negative constructions in declarative sentences are listed here.

3.4.2.1. The negation morpheme -jjo

Negation is regularly marked by the element jjo, which contrasts paradigmatically with the affirmative that is expressed by zero morpheme. Consider an affirmative sentence and its negative counterpart in (A) and (B) respectively.

Example A: affirmative sentence

(a) [ʔáni]PRP$ [ʔítt'o]PRP$ [weeʃ-ú-mm-o] VBD 'I called him.'

(b) neési ni bát'o guull-i-n-ú-mm-o 'We finished our work.'

B. Negative Counterpart

(a) [ʔáni]PRP [ʔítt'o]POS [weeʃ-ú-mm-ó-jjo]VBDN
'I did not call him.'

(b) [neési]PRP [ni]PRP [bat'o]NN [guull-í-n-ú-mm-ó-jjo]VBDN
'We did not finish our work.'

3.4.2.2. The existential negation morpheme beeʔe

Existential sentences, either affirmative or negative, always appear with a dative and locative arguments. The morpheme beeʔe is used to negate the verb of has/have. Compare affirmative existential sentences with their negative equivalents as shown in the following examples.

(a) ka miné-nne mánn-i beéʔe 'No body is in the house.'
(b) ki beet-ína wodán-i beéʔe 'Your son does not have consciousness.'
(c) ʔama-nne-é woʔo-nne-é dʒór-i beéʔe 'There is no either a bad water or a bad mother.'
(d) ʔisé-na maandár-i beéʔe 'She does not have good manner.'

The negative existential verb does not take a subject agreement element. This is, however, not the case with the affirmative equivalent. The followingexampls are negative existentials and affirmative equivalents.

(A) negative existentials

(b) neési ni bat'o guull-í-n-ú-mm-ó-jjo 'We did not finish our work.'
(c) ʔáti kitaabb-oo-llá-jjo 'You are not writing.'

(B) affirmative

(d) ʔítt'i bat'-oó-lli heeʔ-ú-kk-ó-jjo 'He was not working.'
(e) ʔissúwwi kananaaʔ-am-oó-lli heeʔ-am-u-kk-ó-jjo 'They were not writing.'
(f) díʔri maár-i fajjaʔ-oom-ína danaamó-jjo 'Fat meat is not good for health.'

3.4.2.3. The existential negation morpheme beeʔe

Existential sentences, either affirmative or negative, always appear with a dative and locative arguments. The morpheme beeʔe is used to negate the verb of has/have. Compare affirmative existential sentences with their negative equivalents in the folowing examples:

(a) ka miné-nne mánn-i beéʔe 'No body is in the house.'
(b) ki beet-ína wodán-i beéʔe 'Your son does not have consciousness.'

3.5. Interrogatives

Interrogatives can be divided into polar Interrogatives, which elicit 'yes" or "no" answers, and non polar Interrogatives (content interrogatives), which involve content question words. Here is its sub-list:

3.5.1. Polar Interrogatives

Polar interrogatives express questions that can be answered by 'yes' or 'no' and 'may be' or 'I don't know'. To differentiate an interrogative clause from a declarative clause, Hadiyya exhibits two possibilities: One is the distinction in intonation patterns. The sentence in (a) with falling intonation at the end is declarative, while structurally identical (b) with a rise in intonation at the end is interrogative:

(a) ki beʃ-íttʃ-i waattʃam-iínse waar-ú-kk-o. 'Your friend came from Waachamo.'

(b) ki beʃ-íttʃ-i waattʃam-iínse waar-ú-kk-o 'Did your friend come from Waachamo?'

The second possibility is by employing the interrogative particle, -nnihe. Most often this is a verb. Here is the examples for the interrogative sentences in (b) and (d) with the affirmative ones in (a and c).

(a) ku mán-tʃ-i ka diináte ʔuʃeʔ-u-kk-uúlla míne dabaʔl-ú-kk-o
'The man returned to his house driving the cattle.'.'

(b) ku mántʃ-i ka diináte ʔuʃeʔ-u-kk-uúlla míne dabaʔl-u-kk-o-nníhe
'Did the man take the cattle and return back to his house?'

(c) hadíjj-i k'aank'á-nne ʔájji man-tʃ-í-mi leh-u-kk-aá wiʔl-akk-(a)m-o-ó t'idd-akk-(á)m-o
'According to the culture of Hadiyya, when a person of any age passed away, one will cry and be depressed.'

(d) hadíjj-i k'aank'á-nne ʔájji man-tʃ-í-mi leh-u-kk-aá wiʔl-akk-(a)m-o-ó t'idd-akk-(á)m-o-nníhe
'Should there be a cry of sorrow and mourning according to the culture of Hadiyya, when a person of any age passed away?'

Below are given a few more examples of the use of the morpheme -nnihe in polar interrogative clauses.

(a) gos-akk-(á)mi ʔamman-i-í bejj-i-í j-oo-nníhe
(when and where to carry out the abduction (Gosano)?'

(b) gosán-i hadíjj-i k'aank'á-nne los-am-aá-kk-o ʔeeb-akk'-ám-tʃ-i ʔogora-nníhe
'Is Gosano (abduction) a type of marriage which is acceptable in Hadiyya's tradition?'

3.5.2. Non polar interrogatives

Non polar interrogatives are question words.The following are content question words in Hadiyya. Here is the lists of content question words

[ʔájjị̥]WP	‘who’(human, subject)
[ʔájje]WP	‘whom’ (human object)
[máhị̥]WP	‘what’ (non-human, subject)
[máha]WP	‘what’ (non-human, object)
[hánno]WDT	‘where’
[máhina]WDT	‘why’
[hinkíde]WDT	‘how’
[hínkị̥]WP	‘which’ (generic, subject)
[hínka]WP	‘which’ (generic, object)
[hinkiʔammáne]WRB	‘when ‘ (which time)’

These words are often identical to a set of pronouns discussed previously.

The subject question words occur in subject position and object content question words occur in object position. Here is the examples.

(a) ʔísi daaddʒ-iínse máhạ̥ ʔeebb-ó-ʔ-o ‘What did she bring from river?’
(b) ʔísi daaddʒ-iínse wóʔo ʔeebb-ó-ʔ-o ‘She brought water.’
(c) ʔájj-i beebálla leh-ú-kk-o ‘Who died yesterday’
(d) lóbi mán-tʃ-i beebálla leh-ú-kk-o ‘An old man man died yesterday.’
(e) ʔísi ʔájje ʔiitt-i-t-(á)m-o ‘Whom does she like?’
(f) ʔísi keése ʔiitt-i-t-(á)m-o ‘She likes you.’

Predicate interrogative noun phrases too can be fronted and focused. Compare non focused and focused interrigatves in (a) and (b) respectively.

(a) ʔoo mán-tʃ-i ʔajjé-tte? ‘Who is that man?’
(b) ʔajjé-tte ʔoo mán-tʃ-i? ‘Who is that man?

3.5. Relative clauses

“Relative clauses can be prenominal (the clause occurs before the head), postnominal, internally headed (the head occurs within the relative clause), or they may be headless”. With respect to the position, Hadiyya employs prenominal relative clauses.

(a) [geer-oó-kki]RB [beét-i]NN ‘the boy who runs’

(b) [beét-i]NN [geer-oó-kk-o]VBZ ‘The boy ran.’

Relative Clauses use either an NP headed by a noun or reduced agreement marker of the noun-phrase as their head. Since relative clauses are noun modifiers, they occur in the same position as other noun modifiers such as descriptive adjectives, numerals, etc.

(a) lóbi mánn-i ʔítt’i ʔabbaájjo [ʃ-ú-kki] mán-tʃo
‘Old people cursed the person who killed his brother.’

The head of the relative clause in (a) is mántʃo ‘man’. Just like other nominal modifiers this noun is preceded by the relative clause in brackets. All the following **noun phrase** constituents allow relativization: subject, direct object, indirect object and genitive (possessor). Consider illustrative examples:

(a) maaddeéb-I gameʔl-ú-kki mán-tʃ-i waar-ú-kk-o
‘The man who Maaddeebo insulted came.’
(b) maaddeéb-i ʔeeb-ú-kki meent-íttʃi ʔabbaájj-i
‘The man whose sister Maaddeebo married came.’

The head of the relative clause is the common noun mántʃ-i [man-NOM], and is the noun-phrase morpheme -kki in (b). The following examples are further illustrations, contrasting subject and object relativization including non-relativized clauses.

(a) mán-tʃ-i beebálla gereé-ttʃo dur-ú-kk-o
‘A man slaughtered a sheep yesterday.’
(b) gereé-ttʃo dur-ú-kki mán-tʃ-i beebálla waar-ú-kk-o
‘The man who slaughtered a sheep yesterday came.’ (subject relativized)
(c) mán-tʃ-i beebálla bataaʔ-ú-kki gereé-ttʃ-i leh-ú-kk-o
‘The sheep that a man bought yesterday died.’ (object relativized)

“Headless relative clauses are those clauses which themselves refer to the noun that they modify”. Headless relative clauses are recurrently used in Hadiyya. The folowing example demonstrate relative clause with and without head in (a) –(b) respectively.

(a) [ʔabuúlla̱ tʃ’eem-ú-kki] mán-tʃ-i ʔaboʔ-í-nne ʔuuntt’-oó-kk-o
‘Who he does not like to work, begs with his white hair’
(b) [ʔabuúlla̱ tʃ’eem-u-kk-ó-kki] ʔaboʔ-í-nne ʔuunt’-oó-kk-o
‘Who does not like to work, begs even after his hair turned grey.’

3.6. The converb

The converb is a verb form that functions as a clause linking device and does not form a sentence on its own. The converb markers are -aa and -ooʔne, which are affixed to the affirmative and negative verb stems respectively.

(a) gabálạ hig-aá hark'oot-ú-kki man-tʃ-ína ʔuww-akk-(á)mi ʔoogát-i máhạ

'What will be the punishment to be given for the trespass and plough the farm land?'

(b) máti mán-tʃ-i máti land-íttʃo gos-aá ʔeeb-oo-kk-ó-ki mahi-ná-tte

'Why a man abducts a girl and gets married with her?'

In complex sentences containing several converbs, the first converb expresses an action that takes place first, the second converb the next action, etc. Here is an examples:

(a) [ʔángạ]NN [ʔaanʃ-akk'-aá]VBD [hooʃʃóʔo]NN [ʔit-ú-kk-o]VBD

'After he had washed his hands he ate lunch.'

(b) [hooʃʃóʔo]NN [ʔit-aá]VBD [ʔángạ]NN [ʔaanʃ-akk'-ú-kk-o]VBD

''After he had eaten lunch he washed his hands.'

3.7. The Conditional Clauses

A conditional clause is subordinate to a main clause and it is marked by -re and -daʔne. The conditional clause occurs before the main clause. The conditional marker -re is obligatorily preceded by a converb marker, -aa. Consider the use of -re as real and -daʔne as unreal conditional marker in I) and (II) respectively.

(I a) ku land-íttʃo ka ʔeeb-akk'-ám-tʃạ sabb-o-ʔ-aá-re máhạ ʔiss-akk-(á)m-o

'What if the girl refuses such a type of marriage?'

(b) ka k'araárẹ likitʃ'-í-tt-aá-re le-t-oó-jjo

'If you drink this medicine, you won't die.'

(II a) keene mar-ú-mm-í-daʔne ʔub-ú-mm-í-heeʔ-ú-mm-ó-jjo

'If I had gone this way, I wouldn't have fallen.'

(b) ʔi-ína ʔammáni heeʔ-u-dáʔne keésẹ haraʔm-u-m-heeʔ-ú-mm-o

'If I had had enogh time, I would have helped you.'

3.8. Conclusion

In this chapter the parts of speech or word categories of hadiyya language are described to provide clear views on how the parser corpus is obtained from the documentation of hadiyya. The word categories includes :- Noun, Pronoun, Verb, Adjective, Adverb, Cardinals, Conjunctions are stated.

The lights towards syntactic grammer for Hadiyya theoratically provided by the lingutic researchers for Hadiyya language. The research paper provide the documentation-based descriptive grammar with the aim of giving a comprehensive description of Hadiyya language, Basic Linguistic theory of the language as the researcher thought within the methodlogy (Tadesse, 2015). Thus needs additional formalisim of the computitional lingustic syntax analysis system inorder to create natural language understanding system.

The collected corpus cannot be directly used because deep analysis based on context free grammar syntesis of parts of speech is requried as a model as it is described in the next chapter.

CHAPER FOUR

DESIGN OF THE HADIYYA LANGUAGE PARSER/SASPFHL/

4.1. Introduction

The process that maps a sentence to its syntactic structure and logical form is called the parser. It uses knowledge about word and word meanings (the lexicon) and a set of rules defining the legal structures (the grammar) in order to assign a syntactic structure and a logical form to an input sentence . Computational models are useful both for scientific purposes-for exploring the nature of linguistic communication and for practical purposes for enabling effective human-machine communication (Allen, 1995).

In most contemporary grammatical formalisms, the output of parsing is something logically equivalent to a tree, displaying dominance and precedence relations between constituents of a sentence, perhaps with further annotations in the form of attribute-value equations ('features') capturing other aspects of linguistic description. However, there are many different possible linguistic formalisms, and many ways of representing each of them, and hence many different ways of representing the results of parsing. We shall assume here a simple tree representation, and an underlying context-free grammatical (CFG) formalism. However, all of the algorithms decribed here can usually be used for more powerful unification based formalisms, provided these retain a context-free 'backbone', although in these cases their complexity and termination properties may be different (Pulman, 1991).

This chapter is as it comprises of a core domain of this thesis describes how SASPFHL system performed, how the Knowledge base for automatic refernsing for corpus for POS taged dataset for training the machine, the architecture which figure out how the sub components are organized to attain parsing goals. Inaddition to this, the prototypes of SASPFHL system presented at the end of this chapter. Prtotype is a GUI model designed to aid for interacting with SASPFHL, the Hadiyya language parser developed as it is proposed in chapter one methodology section.

4.2. Methodology and Work Flow

This section presents the proposed method employed in this thesis. A Supervised Automatic Sentence Parser for Hadiyya Language (SASPFHL) uses both bottom up and topdown parsing strategy. This strategy employed chart parsing algorithm which uses a dynamic array and list generic collection to store the words, Parts of Speech (POS) tag, Maximum Entropy probabilistic

mapping of result from MaxEtropy model for training the system and predicting unseen word for supporting the production of syntactic tree structure with head rules for a sentence collection. The parsing model (automatically refers to POS tagged sentence collection for this thesis it is called HadiyaaPOS) dynamic array index and memory values. The parser is developed using C# .Net environment. In order to develop the Hadiyya language sentence parsing model we applied various step wise process which involves: a) text preprocessing b) Sentence Parsing C. generate an output.

4.2.1. Text preprocessing

The outside source that SASPFHL requires is the corpora of grammar, lexicon and dectionary saved in text model files, which is requred for automatically traing the system. Thus, this dectionary file is a collection of Hadiyya Dictionary dataset, Grammer is a production rules that guide the parser how to arrange sentences syntactically, it is known as Hadiyya sentence Head rules, POS tag for Hadiyya sentence collection for automatically acqure a set of word with there tag to automatically learn the structure of syntax for syntesis of parsed sentences for raw text input. Without this file, SASPFHL does not have the corpora to reference to.

4.2.2. Parse the sentence in to word list

The parsing of syntax analyzer is done through following rule based algorithm .

1. Enter a sentence .
2. Categorize the sentence using Phrasal category like NP, VP, ADJP, ADVP, PP.
3. Check the phrases of sentences using various tags that are returned by POS tagger. (the tags asociated are noun phrases (N, NP, NNP) and verb phrases (V, VP, IN)).
4. Partition the sentence into NP and VP identified.
5. MaxEntropy model train from identified set features and compute the probability of rule and POS tag and return to extraction algorithm.
6. Parse sentences with computed pattern like NP, NNP, V and VP by matching it against the Grammar head rules.
7. Print syntactic output on the output text area (If all parts of the sentences are parsed correctly then sentence is syntactically correct, else the sentence is syntactically incorrect).

4.2.3. Generate Output

SASPFHL is Hadiyya sentence parsing prototype that generate all complete node with identified sentence starting symbol “S” by using identified boundary symolys (like full stop for declarative sentences, question mark for introgative sentences and exclamation mark for commands). The parts of speech along with a sentence category like NP (Noun Phrase),VP(Verb Phrase),PP(Prepositional

Phrase), ADVP(Adverb Phrase), ADJP(Adjective Phrase) binds tags with words within the fragments(such as: NN(NOUN), VB (Verb), IN (Preposition), CD(Cardinals)).

4.3. Implementation Tool

As discussed in the above sections of this thesis, to develop this systems prototype of the system and algorithms were implemented in Microsoft Visual Studio Professional 2015 version 14.0 Update 2 Integrated Development Environment (IDE) used to execute corpus preprocessing and sentences parsing.

The reason why the researcher uses this IDE because of a numbers of advantages. First, the Visual Studio is well optimized with excellent professional end for building graphical user interface with built in controls. Another reason is, this IDE gives flexibility of importing required library for integrating to your project from another source or third party, for natural language processing, especially for this work sharp Entropy reference library was imported as reference. Moreover, it is optimized for software quality, developer productivity, program portability, and component integration.

Finally, the reason why visual studio selected is good for encoding a symbol and alphabet otherthan the english alphabet. The sentences of Hadiyya languge containg special symbols(like, ʔi meaning my) was excuted without dificulty for pre-processing and representation in computer memory. So that, the expression of special types of script easly automatically compiled by the visual Studio 2015 IDE. The system run on windows platform which is most popular operating system with its ease of user interaction.

4.4. Knowledge about Control

It is not sufficient to have access to rules which define what strings are well-formed and what structures can be built; a parser also needs to have knowledge of how to apply them. It is important to distinguish between control decisions which are taken simply to increase efficiency - like keeping a WFST to save re-computation - and those which affect the nature of the result - like rule-ordering in a top- down, depth-first parser which halts as soon as it finds a solution.

The required knowledge for building natural language grammatical structure understanding system is known as syntactic knowledge. Syntactic knowledge is concerns how words can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of what other phrases (Allen, 1995).

4.5. Hadiyya Parser System Model

The suggested archtecture for parsing the Hadiyya language has two phases: the Preprocessing phase and the analysis phase. The first phase requires a training corpus, extraction features and a set of rules extracted from the learning corpus. The second Post Processing phase implements the learning results from the first phase to achieve parsing. The phases of Hadiyya syntactic parsing approach are illustrated in the following figure:

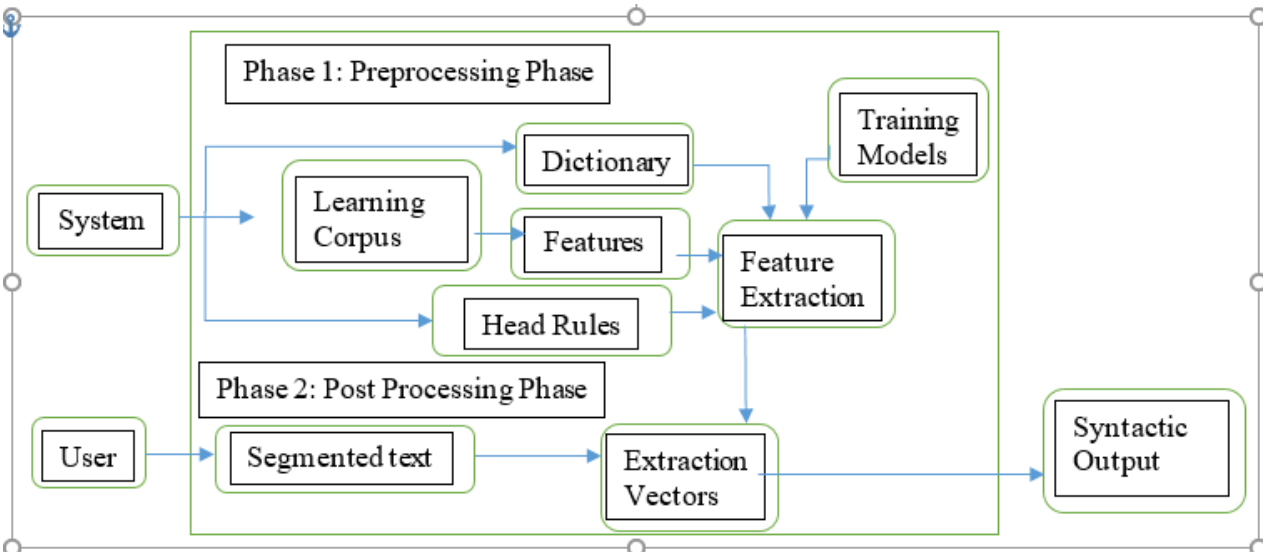


Figure 4. 1 The SASPFHL System Architecture

4.5.1 The Preprocessing phase

The preprocessing phase involves the use of a training corpus, a set of features and rules extracted from the learning corpus analysis in order to train the hybrid approach (the chart parsing with Maximum Entropy classifier). Learning corpus. POS tags consist of coded abbreviations conforming to the scheme of the Penn Treebank, the linguistic corpus developed by the University of Pennsylvania. It is composed of data from standard and modern linguistic sources written in Hadiyya. It comprises 200 sentence texts of Hadiyya language as prepared by the researcher (Tadesse, 2015) and with text preprocessing. In the learning phase, the resercher corpus was used.

Extraction features. These features indicate the information used from the annotated corpus during the training stage, which is the morphological annotation. We classified these features into two classes namely, part of speech features and contextual features: A part of speech (POS) feature specifies the morphological category of the word being processed. (See Appndix D)

A contextual feature indicates the POS of the words in the left vicinity of the word being analyzed. In (Appendix C) for chart based syntactic text structure obtaing algorithm, which explain the steps to extract the different features used to determine syntactic analysis with respect to grammer rule:

Extraction rules. These rules are derived from a deep analysis of the Hadiyya language sentences. They are used to train our system in grouping the sequences of labels that may belong to the same syntactic grouping and thus better define their borders. The combination of features and rules extracted from the training corpus allows allocating each word in the sentence to its most probable syntactic group and thus training our analyser to classify them automatically. These rules have the following structure:

Rule : {M1, M2, M3, M4, M5, M6} Si

Where M1 through M6 represents the morphological category of the words in a given syntactic group and Si represents the syntactic class of the group. A rule may be composed of one, two, three, four or five elements. We extracted rules from the Hadiyya Sentence Collection. Here are some examples of extracted rules and the remaining rules are on Appendix G :

R1 : IN,NN NN
R2 : JJ,IN,JJ ADJP
R3 : NNPS NP
R4 : IN,NN, PRP$ PRP
R5: NN VBD
R6: PRP$

Generation of the extraction vectors. This step aims to annotate each word of a sentence in the learning corpus according to the different extraction features presented above. Extraction rules are also used to identify the syntactic class of the groups of words.

Each group of word is described by a vector called extraction vector. The nominal value for a given feature corresponds to the morphological annotation of the word (adjective, noun, verb, punctuation, etc.) according to features used. This vector is completed by the appropriate syntactic class (NP, VP, PRP...) selected from the syntactic annotations in the Hadiyya corpus. The set of extraction vectors forms an input file.

For the learning stage. At the end of this process, the learning corpus is converted from its original format into a vector format and we obtain a tabular corpus which consists of a set of vectors separated by a return as shown in the following example:

Vector1 : IN,NN,? ,? ,? , PRP

Vector2 : JJ,CC,JJ,?,?JJP

Vector3 : NNPS,?,?,?,?,NP

Learning. This stage uses the previously generated extraction vectors in order to produce equations known as hyperplanes equations. The learning algorithm used in this stage is the chart parsing algorithm. To my knowledge, there is no works using chart parsings for parsing the Hadiyya language.

Since chart parsings are using Well Formed Sub String(WFSS), it uses an array data structures for storage unit. This algorithm generates several hyperplane equations which are used to classify the different word groups according to their appropriate syntactic class (NP, PP, VP, ADJP and ADVP). It is noteworthy that the learning stage is done only once and is only repeated in case we increase the size of the corpus, or change the type of corpus.

This step is performed using 20 % of the Haddiya Sentence collection from the researcher (Tadesse, 2015) and the Open NLP library done with CSharp a Templet of Visual Studio 2015(Version 2.0) was used. This tool takes as input extraction vectors in the form of an ".nbn" file and outputs syntactic text result.

4.5.2 The Post processing Phase

This phase implements the results of the learning phase in order to parse a sentence. The user must provide a segmented sentence as input to the parsing system. This phase proceeds in two steps as follows: Firstly, a pre-processing is applied to the input sentence. Indeed, the system use features and rules to arrange words in groups following the vector format as presented in the learning stage. This pre-processing generates extraction vectors like those generated as input for the learning stage. The only difference is that these vectors do not contain the syntactic class. This information will be calculated by the Maximum Entropy.

Then, the extraction vectors generated in the first step and the hyperplane equations generated in the learning stage are provided as input to the classification module. Indeed, for each vector, we calculate a score using probablistic context Free Grammer value with Maximum Entropy. Each

equa-tion discriminates between two syntactic classes (e.g. POS tag/Phrase type). So every vector will have 0 or 1 scores according to the number of equations. The score and its sign are used to identify the suitable syntactic class for the test vector.

At the end of this stage we obtain a parsed sentence in a tree form or braketed sentences.

4.6. Parsing Hadiyya Noun Phrase

Noun phrases (NPs) are used to refer to things: objects, places, concepts, events, qualities, and so on.A noun phrase in this section represented with NP (Noun Phrase) in the output of a parsed sentence. A noun phrase can be branched as NN which becomes a head on the left and specifiers and modifiers on the right of the phrase as in the following example. The specifiers and modifiers are made of Adjectives, adverbs and numbers in Here is an example how to parse different types of noun phrase:

A) Adjective-Head noun

(a) (S (NP (JJ gundi̱) (NN man-tʃo̱)) (. .))

‘a short man’

(B) Numerals-Head noun

(a) (S(NP (CD lámi̱) (NNS ʔabbaajj-úwwa̱))(. .))

‘two brothers‘

(C) Qunatifier-Head noun

(a) (S (NP (JJ lobakáti̱) (NNS ʔoósо̱)) (. .))

‘many children‘

(D) Demonstrative-Head noun

(a) (S (NP (DT ku=) (NNS meént-i̱)) (. .))

‘these women’

(E) Possessive-Head noun

(a) (S (NP (POS bak’úttʃi) (NN lókkо̱)) (. .))

‘mules leg‘

(F) The relative clause-Head noun

(a) (S (ADVP (RB foóre̱)) (NP (NP (NN mán-tʃ-i̱)) (VP (VB ʃ-ú-kki̱) (ADJP (JJ heemáttʃi̱) (NN gereé-ttʃo̱))) (. .))

First, the person who killed will deliver a black sheep.

Parsing the personal pronuons. Example:

(c) (S (NP (PRP$ ʔítt'i) (NN baár-i̱)) (. .))

'his ox'

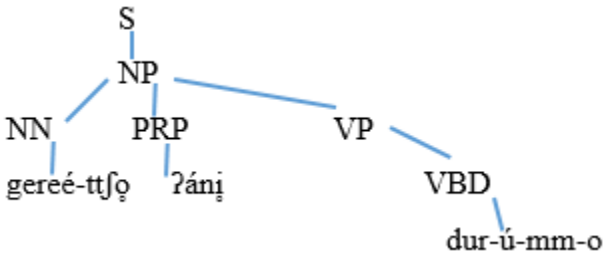
4.7. Parsing Hadiyya VerbPhrase

In Hadiyya grammar the most frequent word order in intransitive and transitive sentences is respectively, Subject –Verb(SV) and Subject Object Verb(SOV). Example:

(A). (S (NP (NN gereé-ttʃo̱) (PRP ʔáni̱)) (VP (VBD dur-ú-mm-o)) (. .)))

'I slaughtered a sheep.'

Or the tree like euvalent production for this senetence is:



In command sentences involving the dative, the Subject-Dative complement-the Object-Verb order is most frequently used: Here is an example:

(a) (S (NP (PRP ʔáti̱) (PRP ʔisé-na (NN diinátẹ̱)) (VP (VB ʔúww-e)) (. .))

'You! Give her money.'

4.8. Parsing of Punctuation

The white space and the comma are used as a delimiter to identify individual words in a sentence written in Hadiyya sentence. Moreover, the following marks are also used as marks which sign the end of a sentence: the full stop (.) with declarative sentence, the question mark (?) with interrogative sentences and the exclamation mark (!) with command sentences. During parsing these punctuation marks are used for telling the parser the end of statement. For exampe, the result of parsing end of the sentence will produce the following structues !/., ./. and ?/.

4.9. **The Prototype of System (SASPFHL)**

With in this section the parsing system pototype is prsented. For the purpose of this research the prototype system which we called SASPFHL (Supervised Automatic Sentence Parser for Hadiyya Language) is developed for demonistrating entry of Hadiyya language sentence user input and production of syntactic output on the user interface when the user pressing on Parse commanding

button. The prototype of the parsing system for Hadiyya language is developed using Microsoft Visual Studio 2015 proffesional integrated Development Environment with C#(CSharp) .NET Programming templet.

The model is proved applications intended as aids to human activities rather than as free-standing agents. For this reason it is important that the interface between human and computer be user-friendly, in the sense of being easy to learn, use and understand. The history of computing is, in part, a story of increasingly friendly interfaces (Matthews, 1998). The following figure 4.2. shows prototype of SASPFHL system.

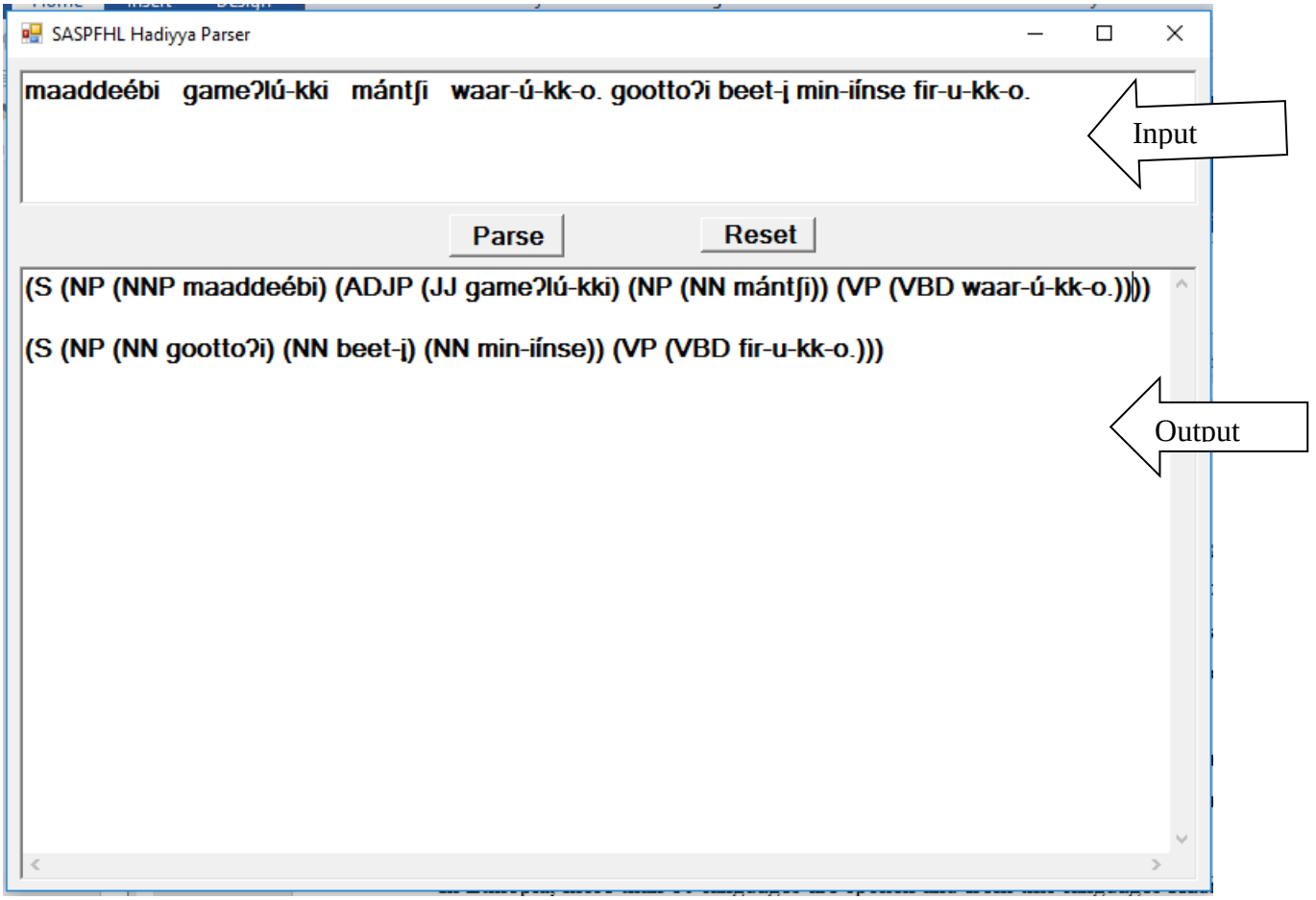


Figure 4. 2 The Prototype of SASPFHL

Description for SASPFHL system GUI is thought as follows:

Parse Button: Handling user click event for commanding a SASPGHAL system for returning syntactically proved tree featured sentence like output displaying with bracketed features.

Reset Button: remove user entry from an input text, in order to re-enter another text as requirement for parsing.

Note: Button texts are written in English word on the Graphical User Interface(GUI) in figure 4.8. is due to ambiguous to find equivalent word in Hadiyya language.

4.10. Conclusion

This chapter discuss about deep analysis of selective examples of sentences desribed in chapter three part 3.4.1 and 3.4.2. which shows the design of syntactic analysis based on Noun phrase and Verb phrase, Adverb phrase, Prepositional Phrase and Adjective Phrase. The other types of phrases are included in expermental data sets. The sentence was categorized in to declarative sentence, exclamatory sentence, and introgative sentences used for syntactic analysis.

The next chapter dicuses about the design of experment result for the Hadiyya Language sentence parsing system, the possible suggested architecture is design and extraction algorithms are stated. Thus, the next chapter exibits the actual expermental result of the parser.

CHAPTER FIVE

EXPERMENT RESULT AND DISCUSION

5.1. Introduction

In this, section of the thesis experiments conducted and results obtained have been described. Discussion on the process of identification of the error using the SASPFHL prototype elaborated with error decryption. The same set of error analysis method is followed for Training Data Set and Test Data Set. Ten times in number, cross validation has been performed in all cases.

In order to examine effects of various preprocessing steps (POS-tagging, lemmatization and syntactic parsing) on system performance more prototypes had been made. The sentences for manually processing (Assigning POS tag for words in a Hadiyya sentence) Hadiyya corpus required for preprocessing was taken from Phd thesis with the title “Documentation and Description of Hadiyya” by (Tadesse, 2015). The sentences are not complete with the description and its punctuation. It also, without expressive parts of speech tags for the words that constitute a sentence. Thus, the corpus was not anotated.

In the system evaluation section within this chapter, syntactically parsed output ratio which is a parser accuracy evaluation result is discussed.

5.2. The Sample Text and the Manual Parsing Process

The dataset used for parser training set and test set is acquired from the unpublished Phd, thesis with the title “Hadiyya Documentation and Description “ sentences morphologically syntesisized collection conducted by (Tadesse, 2015), as he suggested for parser developers the sentences are some what refined to make similarity with dectionary words. The total sentence is numbered to 549 of the Hadiyya text most common simple and conditional statements used for the purpose of the experiment conducted.

This sample was selected since it was prepared with the aim of further study for syntactic parsers of the Haddiya grammar and, thus, it is comprehensive and has also a direct relevance to this study as it apears with english transilation.

The sample text was not annotated with grammatical categories to be readyly available for the parser direct use and it was also laborious, expensive and time consuming to manually parse a large corpus. In addition, preparation of the Lexicon and Grammar Rules for large corpus requires

expense and time apart from the fact that it is demanding. The sample text was given to one linguists (native speaker of Hadiyya). There were some differences between the computitional parser system and the lingustic parsing. That output was used for the experiment.

5.3 Preparing the Sample Data for the Experiment

For the perpouse of doing an experment we do have totally 549 sentence unlebeled corpus which has diffirent sentence types like declarative sentences, introgative sentences and exclamatory sentences obtained from the research paper based on the researcher perorir knowledge and lingustics advisor. For extracting the features of a sentence asigned a head phrase structure to the traning set and the test set. Thus, the parts of speechs are prepared to labeled a corpus (see Appndex G), in order to evaluate the parser output correctness by comparing with manualy parsed sentences.

Having more amounts of training set for the parsing system gives a clue for pattern extraction to formulate syntax of unseen sentences. From 549 sentences purposively taken with three types of sentences which are the declarative, introgrative and exclamatory sentences included in both training set and test set. The sample data was divided into two sets; the training set (taken from the first page of the sampled text until it reaches 70.8 % [389 nodes in the text] of the total corpus and the testing set [29.2%] left from the text). A sample of the text taken is found in the appendix B.
During the preparation of the experiment, the punctuation marks necessary to indicate end of the sentence (!,. , ?) added in text. The end of sentence symbols are used for the parser to detect a sentence boundary from free text. Another

5.4 Evaluation Procedures

For the purpose of this research, only the percentages of correct Parse assignment to measure the performance of the Parser.
During the experiment time the following steps ware performed to evaluate the performance of the Parser.

A. The Parser was first trained on the training set obtained earlier from the texts. The Parsing algorithm for POS extraction, the tree construction algorithms and morphological(feature extraction algorith) were the parser is used to excute.
B. The result obtained on the training set was then evaluated by comparing it with manually Parsed text used as a training set. Aiming to improve the Parser accuracy, causes of error were identified and corrected and step one was repeated again. These two processes repeated until

the result obtained was found to be efficient on the training set to see how well the Parser performs on the training set.

C. Evaluate by entering test set to the parser input Text Area. To persive whether the Parsing system was traind to address the ambigutiy for predicting new word is attaind and testing system algorithm efficency of its performance. The test sets should be sentences not same but may be similar to training set.

D. The output of the Parser on the test set was then compared with the manually Parsed sentences of the same corpus. The steps C and D will continue upto the best efficency performance level.

5.5. Evaluation Metrics

In this thesis, the testing was done mainly of an intrinsic, black-box and crosscheck evaluation. The standard techniques for evaluating parsers and grammars are called the PARSEVAL measure. We can measue using this techniques the percentage of parser accurecy interms of precision, recall and f-measure (Daniel Jurafsky, 2009).

Precision and recall are concepts first widely used in evaluation of sentence parsing systems. Precision and recall can be used in evaluating information extraction as well. In this case, precision is the number of items correctly labeled as belonging to the class of interest (true positives) divided by the total number of items labeled correctly or incorrectly as belonging to that class (true positives + false positives). Recall, by contrast, is the number of items correctly labeled items (true positives) divided by the total number of items that actually belong to the class which includes items not correctly identified as belonging to that class (false negatives).

Mathematical expression for Precision P and Recall R respectively as follows,

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP+FN}$$

Precision and recall are inversely related. When precision is increases, the recall decreases. For example, in syntax analysis the parsing system return all sentence parse structure including missed phrase tag, if the output of the system prototype is result without false negative and all sentences are extracted correctly and then we get a 100% recall, whereas the precision decreases significantly as the user input sentences are not extracted correctly.

Usually, precision and recall are combined to give one value called the f measure. The f-measure is the weighted harmonic mean of precision and recall to measure accurecy.

Mathematically,

$$F = \frac{2PR}{P + R}$$

F measure is the harmonic mean of precision and recall, where both measures are given equal importance.

5.6. Expermet on Data Sets

The performance of our system evaluated before correction and after correction. The results of system prototype are elaborated in the next section. The researcher do obtaining the output accuracy relevance for the Hadiyya language sentence through crosschecking the system result with parsed sentence. The system Graphical User Interface (GUI) has two text area. One text area used to enter a user input and another text area is used to display the system sentence parse result. Here in the figure below showed us how the sample test set sentences are identified whether correctly parsed or not.

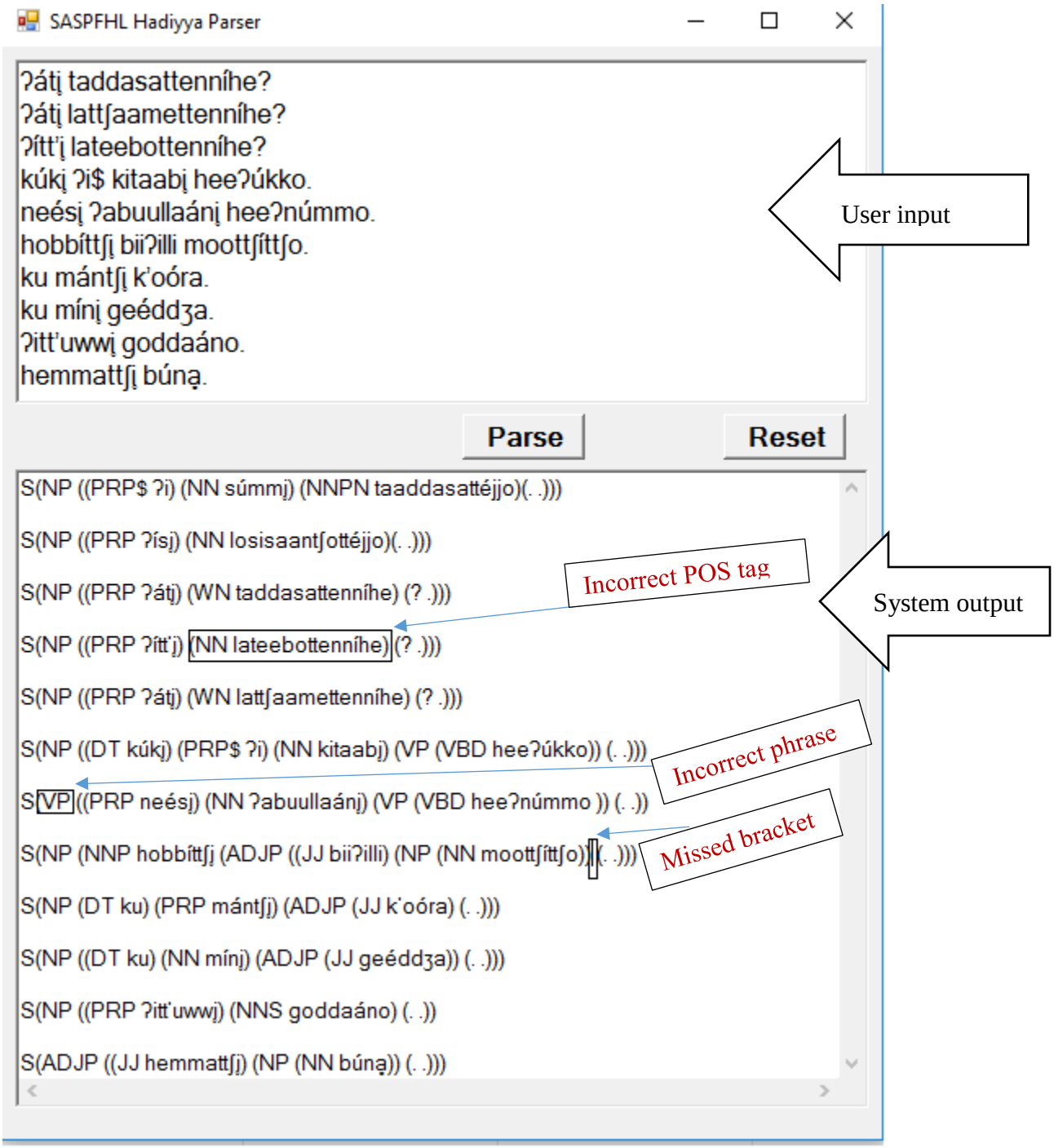


Figure 5. 1 Sample Error identification on Test Data Set

The result for each correct and incorrectly parsed sentences counted by the researcher for evaluating the accuracy of the Hadiyya language sentences parsing system. Three types of error (assigning incorrect POS tag to a word, incorrect phrase tag and rearmost one is missing bracket) is exhibited on the system result as shown in the Figure 5.1. In the first error is from the GUI of the snapshot the forth sentence is containing incorrectly tagged word 'latt?aamettenníhe' as Noun (NN) but the correct POS is Which determiner noun (WN). The second error is on the seventh sentence is happen

when the pattern identification model see the verb on the sentence to assign Verb Phrase (VP) tag. In the sentence verb phrase (VP) is preceded by Noun Phrase (NP) so this sentence is incorrectly parsed. The list occurring error is missed bracket the reason is that sentence-parsing system assume the words are having the same tag set.

5.7. Result on the Training Set and Test Set

Errors occurred at testing time was incorrect Tags assigned by linguistics and results from unsupervised model it try to assign few wrong tags when it follows headed rules pattern. For example,

S(NP((DT ki) (NN súmm-i̱) (VP (VB ʔajjé-tte)) (? .)))

Determiner noun verb

But the correct sentence word Parts of speech tag is the following:

S(NP(PRP (ki) (NN súmm-i̱) VP(VB(ʔajjé-tte)) (? .)))

pronoun noun verb

The parsing results of the sentences are crosschecked with manually synthesized corpus. Three types of parsed sentence is identified. These are: true posetive(correct syntactic output sentence), False Positive (sytactically structured but incorrectly labeled) and false negative(incorrectly parsed totall sentences including incorrectly labeled and unlabeld).

The result obtained when the Parser was trained and run on the same data, TrainingSet, is shown in table 5.1.

Data/set	***No of sentences***	***# of Correctly parsed sentences***	***Incorectly parsed senteces***
TrainingSet	389	120	169

Table 5. 1 Parsing result before making no error correction

The expermental result for parser acurcy measurement before correction on training set is as follows:

Data/set	***Recall %***	***Precision %***	***F-Masure %***
TrainingSet	30.8	40.0	34.8

Table 5. 2 Accuracy of Parsing result before making no error correction

Data/set	***No of sentences***	***# of Correctly parsed sentences***	***Incorectly parsed senteces***
TrainingSet	389	230	159

Table 5. 3 Parsing result after making consecutive error correction

The expermental result for parser acurcy measurement before correction on training set is as follows:

Data/set	*Recall %*	*Precision %*	*F-Masure %*
TrainingSet	59.1	67.2	62.8

Table 5. 4 Accuracy of Parsing result after making error correction

Data/set	*No of sentences*	*# of Correctly parsed sentences*	*Incorectly parsed senteces*
Test Set	160	45	115

Table 5. 5 Parsing result before making no error correction

The expermental result for parser acurcy measurement before correction on test set is as follows:

Data/set	*Recall %*	*Precision %*	*F-Masure %*
TrainingSet	28.1	36.0	31.5

Table 5. 6 The parser accuracy result before making no error correction

Data/set	*No of sentences*	*# of Correctly parsed sentences*	*Incorectly parsed senteces*
Test Set	160	103	57

Table 5. 7 Parsing result after making error correction on test set

The expermental result for parser acurcy measurement before correction on training set is as follows:

Data/set	*Recall %*	*Precision %*	*F-Masure %*
TrainingSet	64.3	73.0	68.3

Table 5. 8 The parser accuracy result after making error correction

The experment was conducted up to ten times in number reprocessing to obtain good result. The result of this parser is cross checked againest manually annotated data which was colaboratly done by the researcher and lingustic expert for this research. Generally, the result of experiment showed that, parsing system can still be improved. Although this algorithm shows good result on precision, that is 73.0%, a cumulative recall is lower at 64.3% and F-measure was 68.3%.

The effectiveness evaluation of a system inidicate that lower recall and higher precision totally acceptable. Thus, the parser is good in sytactic analysis accurecy because the output result relevent sentence grammatical structure for the user input.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

6.1. Conclusions

Supervised learning is the dominant methodology in machine learning. Supervised learning techniques are more powerful than unsupervised techniques because the availability of labelled training data provides clear criteria for model. The hybid approach(Supervised and Unsupervised Machine Learning) for parsing the hadiyya language text is applied succssfuly to succeed morethan 50% accuracy.

An automatic Hadiyya Language sentence Parsing system (SASPFHL) prototype is built with Visual Studio .Net 2015 Integrated Development Environment(IDE) in CSharp programming Language. The parser is based on supervised rule based chart parsing algorithsm supported by MaximumEntrpoy statistical model for the computation of Context Free Grammar(CFG).

In this thesis a modified version of the chart algorithm was applied to a corpus that was manually annotated with a skeletal syntactic bracketing which confirms to the Penn Treebank POS Tags (Daniel Jurafsky, 2009). In the experiment the Input /Output algorithm was modified to consider only PCFG rules which computed with Maximum Entropy general technique for estimating probability distributions from data that did not violate the skeletal bracketing of the corpus. For testing SASPFHL ten times in number cross validation has been performed in all cases. From the experiment accuracy, results we have obtained Precision 73.0 % and Recall owe to 64.3 % on test data set.

An integrated hybrid (rule based and unsupervised) techniques for Hadiyya language parser has been proposed in that handles syntax as well as POS identification from hadiyya language sentences and ease the task of handling syntactic issues in machine translation. This sentence parsing system is based on analyzing an input sentence and converting into a grammar based structural representation. A chart parsing methodology for Hadiyya natural language sentences is proposed and shows how phrase structure rules can be implemented by top-down and bottom-up parsing strategy to parse simple and compound sentences of Hadiyya nation. Absence of contextual word for some candidates in a dataset is the major problem, since the algorithm extracts candidate texts based on this feature.

6.2 Recommendations

This study is not a fullflaged system of parsing a Hadiyya sentances as SASPFHL deals with the syntactic parsing of simple and conditional clause sentences for the Hadiyya language. The following recommendations are forwareded for feature work:-

1. This study directed to hybrid (supervised and unsupervised) machine learning technique. Thus, it is an opportunity to do a research with unsupervised machine learning approach.
2. The scolars can do a sudy on complex compound sentence sytactic parsing of the hadiyya language with context-dependency grammer.
3. The efficency of different algorithms are not tested with this study, by using supervised machine learning approach.
4. The SASPFHL is not concerned with semantic parsing of the Hadiyya language and Grammar. The researcher can take time to study to develop semantic parsing.
5. Using this research result as an input a number of word processing systems may produced. Noun Phrase recognition, conceptual parsing, word sense disambiguation and machine translations like grammer checker are other possible future research areas.
6. The size of the corpus used for the research is important to realize highly efficent automatic machine translation of Natural language processing. The size limitations of the corpus affect the accuracy of parser evaluation. By increasing the corpus size of Hadiyya language it is possible to train a parsing system and compare withthe obtained result.

References

Alexander Clark, C. F. (2010). *The Handbook of Computational Linguistics and Natural Language Processing.* United Kingdom: Wiley-Blackwel.

Allen, J. (1995). *Natural Language Understanding* (2nd ed.). California : The Benjamin/Cummings Publishing Company, Inc.

Babarczy, A., Gábor, B., Hamp, G., Kárpáti, A., Rung, A., & Szakadát, I. (2005). *A Rule-based Sentence Parser for Hungarian.* Hungary: Budapest University of Technology and Economics.

Bergsma, S. (Fall 2010). *Large-Scale Semi-Supervised Learning for Natural Language Processing.* Edmonton, Alberta: University of Alberta.

Damerau, N. I. (2010). *Handbook of Natural Language Processing, Second Edition.* United States of America: Taylor and Francis Group, LLC Chapman & Hall/CRC.

Daniel Jurafsky, J. H. (2009). *Speech and Language Processing* (2nd ed.). New Jersey : Pearson Education, Inc.

Daniel, J. J., & Martin. (2009). *Speech and Language Processing: An introduction to natural language processing.* Mexico: Pearson Education.

Dick Grune, C. J. (2008). *Parsing techniques: a practical guide* (2nd ed.). (F. B. David Gries, Ed.) Amsterdam, Netherland: Springer.

Diriba, M. (2002). *Automatic Sentence Parser for Oromo Language using Supervised Learning Technique.* Addis Ababa, Ethiopia: School of Graduate Studies school of information studies for Africa, Addis Ababa University.

Dumessa, G. A. (2011). Lexicographic implementation in Ethiopia: The case of three dictionaries published since 1995. *Academic Journals*, 3.

Engvall, O. H. (December 2005). *Internet for Everyone in African GSM Networks.* Stockholm: Scanbi-Invest HB.

Goldberg, Y. (2011). *Automatic Syntactic Processing of Modern Hebrew.* Beer-Sheva: Ben-Gurion University of the Negev.

Harry Bunt, J. C. (2005). *New Developments in Parsing Technology.* (J. C. Harry Bunt, Ed.) United States of America: Springer.

Hian Ann Ting, S. S. (1999). *United States of America Patent No. 5,930,746.*

Hócza, A. (2008). *Automatic Syntactic Parsing of the Hungarian Language Appliyng Rule-based Machine Learning Methods,Summary of the Ph.D. Thesis.* Szeged: Doctoral School in Mathematics and Computer Science of the Faculty of Science of the University of Szeged.

Jones Karen, S. K.-J.-J. (1985). *Automatic Natural Language Parsing.* New York: E. Horwood_ Halsted Press.

Karen Sparck-Jones, Y. W. (1985). *Automatic Natural Language Parsing* . Chichester : Ellis Horwood Limited .

Matthews, C. (1998). *An Introduction to Natural Language Processing Through Prolog* . New York : Wesley Longman Inc.

Mishago, T. (2005, june 15). *Notes on Phonetics - 'A Little bit of Hadiyya Grammar'.* Retrieved from Folklore of the Hadiyya: www.sethiopia.org/phoneticnotes.pdf

Monika T. MakwanaȦ, D. C. (2015). *Survey:Natural Language Parsing For Indian Languages* . Nadiad, India: Department of Information Technology, Dharmsinh Desai University.

Potemkin S, B. (1998). *Unsupervised Parsing of the Russian Sentence.* Moscow: Philological faculty of the Moscow State University.

Pulman, S. G. (1991). *Basic Parsing Techniques: an introductory survey.* Cambridge: University of Cambridge Computer Laboratory, and SRI International.

Sim, R. J. (1989). *Predicate Conjoining in Hadiyya: A Head-Driven PS Grammar.* Edinburgh: Department of Lingustics, University of Edinburgh.

Stanford. (2016, Feburary 2). *The Stanford NLP (Natural Language Processing) Group*. Retrieved from Stanford Natural Language Processing: http://nlp.stanford.edu/software/lex-parser.shtml

Tadesse, S. G. (2015). *Documentation and Description of Hadiyya (A Highland East Cushitic Language of Ethiopia), PhD Thesis.* Addis Abeba, Ethiopia: Department of Linguistics and Philology Addis Ababa University.

Tomita, M. (1991). *Current Issues in Parsing Technology.* New York : Springer Science+Business Media .

Westfall, L. (2009). *Sampling Methods. The Certified Software Quality Engineer Handbook* . United States: The Westfall Team.

APPENDICES

Appendix A: Training data Set (Hadiyya corpus)

suunk'-u-mm-o. suunk'i-n-u-mm-o. suunk'i-t-í-tt-o. díss-e. fír-e. guull-e. mar-e. mass-e. waar-e. ʔafuur-e. ʔuull-e. diss-ehe. fill-ehe. guull-ehe. mall-ehe. mass-ehe. waall-ehe. ʔafuull-ehe. ʔuull-ehe. ʔuww-i-n-oo-mm-o. waattʃ-oo-mm-ulla. waattʃ-i-n-oo-mm-ulla. waattʃ-oo-lla. waattʃ-i-n-oo-mm-ulla-jjo. mar-u-kk-o. mar-am-u-kk-o. mass-i-t-í-tt-o. diss-i-n-u-mm-o. dooʔl-u-mm-o. dooʔl-aa-mm-o. ʔuunt'-u-kk-o. ʔuunt'-aa-kk-o. holl-i-n-u-mm-o. holl-i-n-oo-mm-o. ʔaniˌ ʔuull-oo-mm-o. neeseˌ ʔuull-i-n-u-mm-o. bat'-aa bat'-aa hoog-u-kk-o. guull-aa-mm-o. ka=diinateˌ ʔitt'e-na mass-i-t-aa ʔuww-e. waasąˌ ʔit-aa ʔadoˌ ʔag-aa bat'o mar-u-mm-o. beetoo mantʃoo mooʔ-u-mm-o. ʔeesee keesee weeʃ-u-kk-o. ʔisee ʔísi ʔaroʔoo meeraˌ -nne mooʔ- u-mmo. waattʃamoo gimbittʃoo mar-u-kk-o. mar-ona. bagad-í-nne k'as-e. beet-iˌ waar-u-kk-o min-iínse fír-e. saajj-ína hít'e ʔuww-e. ʔaniˌ gootʃ'-ína k'ulfąˌ bitaaʔ-u-mmo. ku=beet-iˌ min-iínse fir-u-kk-o. ʔoo=fellakk-íttʃ-iˌ hít'eˌ ʔit-u-kk-o. ʔoo=ʔoos-iˌ t'orbeʔeˌ lell-am-oo-lla. matiˌ beet-iˌ waar-u-kk-o. matiˌ hall-íttʃ-iˌ leh-u-kk-o. ʔítt'iˌ wiʃ-íttʃoˌ holl-u-kkiˌ beetoˌ sog-u-kk-o. kukiˌ ʃaana-nníhe ʃaana-jjo gudeʔe. ku=mantʃ-iˌ biiʔli-ʔuull-iínse waar-u-kk-o. miʃ-íttʃ-iˌ hakk'-íttʃ-iínse ʔub-u-kk-o. ʔítt'iˌ waattʃam-iínse kaballa waar-u-kk-o. ʔítt'iˌ zaraaʔm-iínse ʔub-u-kk-o. ʔangąˌ ʔanʃ-akk'-imm-ínne dʒabb-iínse gant-ona. ʔaniˌ waattʃamo mar-imm-iínse gat-oo-mm-o-jjo. ʔaniˌ ʔítt'oˌ leh-iínse gat-is-u-mm-o. gimbíttʃ-iˌ waattʃam-iínse ʔiibb-aalla. ku=man-tʃ-iˌ ka=mantʃ-iínse k'eeraaʔla. ku=mooll-iˌ ʔee=mooll-iínse duta. ku=tʃ'iil-íttʃ-iˌ laar-imm-iínse bagaaniˌ mulli luwwąˌ beet-iˌ ʔaraʔa-nne ʔiinseʔ-u-kk-o. dak'ajje-nne lobakatiˌ ʔoos-iˌ lall-am-oo-lla. ʔaraki ʔoos-iˌ biira-nne lall-am-oo-lla. ʔítt'iˌ duuna-nne hakk'ąˌ kaas -u-kk-o. ʔissuwwiˌ kaameʔe-nne ʔafuur-am-u-kk-o. ʔissuwwiˌ kaameʔi woro-nne ʔafuur-am-u-kk-o. maaddeeb-iˌ sasi saata-nne ʔaf-oo-kk-o. maaroʔo-nne waar-e. ʔaniˌ dara-nne waar-oo-mm-o. man-tʃ-iˌ waar-akk-o-ʔ-o. meent-íttʃo waar-akk-o-ʔ-o. ʔaniˌ los-aan-tʃo. [ʔaniˌ] ʔit-u-mm-o. wiʃ-íttʃ-iˌ ʔítt'oˌ holl-u-kk-o. ʔaniˌ keeseˌ laʔ-oo-mm-o-jjo. ʔíssiˌ kíʔneˌ weeʃ-akk-(a)m-ulla. ʔítt'iˌ ʔíseˌ haraʔm-u-kk-o. ʔítt'iˌ neeseˌ los-is-u-kk-o. gam-aan-tʃ-iˌ ʔitt'-iínse diinateˌ gammeʔ-u-kk-o. ku=ʔasse-tʃ-iˌ ʔitt'uww-iínse ʔasseʔ-am-u-kk-o. ʔítt'iˌ niinn-iínse diinateˌ lik'aajjąˌ ʔaaʔ-u-kk-o. ʔatiˌ ʔiin-iínse k'oora. ʔissuwwiˌ niinn-iínse gitann-uwwa. ʔís'e kiinn-iínse danaamo-tte. ʔítt'iˌ ʔitt'i-gagąˌ t'aga ʔl-u-kk-o. ʔísi ʔisi gagąˌ t'aga ʔl-i-t-oʔo. ʔissuwwiˌ ʔissuwwi gagąˌ t'aga ʔl-am-u-kk-o. ʔísi ʔisi gagi harattʃoˌ ʔiikk'-o-ʔ-o. ʔíssiˌ ʔissi gag-iˌ waar-akk-o-ʔ-o. ki beet-iˌ ʔajje-nne waattʃamo mar-u-kk-o. kukiˌ ʔajjiˌ mine. ʔajjiˌ hakk'-íttʃoˌ mur-u-kk-o. ʔíssiˌ ʔajjeˌ haraʔm-akk-o-ʔ-o. ʔítt'iˌ ʔajje-na diinateˌ ʔasseʔ-u-kk-o. ki beʃ-íttʃ-iˌ ʔajje-tte. ʔajjiˌ hakk'-íttʃoˌ mur-u-kk-o. ʔajjiˌ hakk'-íttʃoˌ mur-u-kk-o. ki beʃ-uww-iˌ ʔajj-aamo. ʔajj-aam-iˌ hakk'-íttʃoˌ mur-am-u-kk-o. kukiˌ mahi korshoʔo. ʔítt'iˌ mahąˌ mooʔ-u-kk-o. ʔítt'iˌ land-íttʃo-na k'ootąˌ mahąˌ ʔuww-u-kk-o. ʔítt'iˌ hakk'-íttʃoˌ mahí-nne mur-u-kk-o. ʔatiˌ mah-ína wiʔl-i-t-oo-lla. ʔítt'iˌ mah-ína dooma mar-u-kk-o. hínkiˌ land-íttʃo lobakata danaamo-tte. hínkiˌ ʔoos-i torbeʔeˌ lall-am-oo-lla. ki beʃ-íttʃ-iˌ hínkąˌ land-íttʃoˌ ʔeeb-u-kk-o. ʔatiˌ hurbaatąˌ hinkí-de sar-t-í-tt-o. ki heettʃ-iˌ hinki-de-tte. ki ʔaroʔ-iˌ hinki-de-tte. ʔatiˌ hiki-de-tte. dukkat-iˌ hanno-nne bat'-oo-lla. ki beet-iˌ hanno-nne-tte. bat'-ímmąˌ sabb -í-ti mantʃ-i ʔate-tte-nníhe. daanettʃoˌ bagad-í-nne ʃi-n-u-mmiˌ mann-iˌ neese-tte. googoˌ bat'-am-u-kkiˌ mann-ína diinateˌ mik'-am-u-kk-o-jjo. ka=woʃʃąˌ mattʃ'ees-u-kkiˌ man-tʃ-iˌ lobakata nadad-u-kk-o. leh-u-kkiˌ hall-íttʃ-iˌ gotąˌ badd-oo-jjo. wattʃam-iínse waar-u-mmiˌ beet-iˌ ʔane-tte. kaballa ʔaraki googo taakkeʔ-n-u-mm-o. neeseˌ waaʔąˌ ʔuunt'-i-n-u-mm-o. ni beʃ-íttʃ-iˌ wattʃam-iínse kaballa dara waar-u-kk-o. ʔaniˌ beeballa kojj-ína bunąˌ kaas-u-mm-o. dooʔl-u-mm-o-jjo. ʔuunt'-u-kk-o-jjo. waattʃ-am-oo-kk-o. waattʃ-akk-(a)m-o. ka=bat'oˌ ʔonsoodo guull-i-t-oo-tt-o. maaroʔo mar-oo-mm-o. soodo waar-oo-kk-o. daridara waar-oo-kk-o. matmati ʔammane gaʔn-oo-kk-o. hundi ʔammane daaddʒe-nne waattʃ-i-n-oo-mm-o. gos-aniˌ bikk-ína ʔaaloʔ-ísa kur-t-aa-tt-o. ʔabuulląˌ tʃ'eem-u-kkiˌ man-tʃ-iˌ ʔaboʔ-í-nne. ʔantabaa-kíttʃoˌ gat-is-eena fella-kkíttʃoˌ kitim-u-kk-o. ʔítt'iˌ soodo waar-oo-kk-o. ʔaniˌ ʔítt'oˌ mooʔ-u-mm-o-jjo. ʔantabaaʔi geer-ímm-iˌ gatt'-iínse hig-oo-jjo. ka=ʔeeb-akk'-am-tʃ-uwwi ʔannann-aatoˌ kur-t-oo-nníhe. ki bet-iˌ guull-u-kk-o-nníhe.

waattʃam-iínse waar-u-kk-o-nníhe. ʔatị ʔítt’ọ ʔiitt-i-t-oo-jjo-nníhe. ʔissuwwị kaballa waar-am-oo-kk-o-jjo-nníhe. hakk’ạ kaass -ehe. ʔooso míne ʔaagg-ehe. ka=bat’ọ guull-ona. ka=bat’ọ guull-i-t-ona. ka=bat’ọ guull-am-ona. ka=bat’ọ guull-akk-ona. ka=bat’ọ guull-i-n-ona. ka=bat’ọ guull-oone. ka=bat’ọ guull-i-t-oone. ka=bat’ọ guull-am-oone. ka=bat’ọ guull-akk-oone. ka=bat’ọ guull-i-n-oone. buttʃạ ʔít-e. t’umma gar-e. t’umma hoss-ehe. waaʔ-ị liʔ-is-ona. waaʔ-ị neesẹ t’umm-í-nne ʔed-ona. waaʔ-ị neesẹ t’umm-í-nne dabar-ona. ʔatị waar-t-i-tt-i-daʔne ʔitt’-ị waattʃamọ mar-oo-jjo. ʔee=kíde hig-u-kkị-daʔne ʔub-am-heeʔ-oo-jjo. u=man-tʃ-ị k’adaalli geree-ttʃọ dur-u-kk-o. k’adaallị geree-ttʃ-ị dur-am-u-kk-o. ʔanị dooma-nne hakk’ạ ʔiik’-u-mm-o. hakk’-ị ʔiik’-am-u-kk-o. ʔítt’ị woʔọ daaddʒ-iínse ʔinkiir-u-kk-o. woʔ-ị ʔinkiir-am-u-kk-o. ʔítt’ị bat’ọ lohị saata-nne guull-u-kk-o. bat’-ị lohị saata-nne beedd-u-kk-o. tʃ’iil-íttʃ-ị ʔuull-u-kk-o. ʔítt’ị tʃ’iil-íttʃọ ʔuull-is-u-kk-o. dukkat-ị lap’-u-kk-o. ʔítt’ị dukkatọ lap’-is-u-kk-o. beet-ị woʔo ʔag-u-kk-o. dukkat-ị beetọ woʔọ ʔag-is-u-kk-o. makkeeb-ị hooʃʃoʔọ ʔit-u-k-ko. makkeeb-ị beetọ hooʃʃoʔọ ʔit-is-u-kk-o. makkeeb-ị beetọ hooʃʃoʔọ ʔit-isiis-u-kk-o. ʔare-e ʔaroʔí-ị suunk’-amam-u-kk-o. ʔoos-ị t’oreet’-amam-u-kk-o. ʔítt’ị mass-akk’-u-kk-o. ʔísị lobakata k’oora-tte. gimbíttʃ-ị waattʃam-iínse ʔiibb-aalla. dʒawwoor-ị niinn-iínse k’oora. ʔanị ʔitt’-iínse gunda. ʔísị ʔítt’ọ ʔitt’-iínse ʔiitt-i-t-(a)m-o. ʔotoor-ị dʒaadʒur-íinse kiid-aamo. ku=ball-uww-ị geeddʒ-uwwa. ku=ʔoos-ị k’ooll-uwwa. ʔitt’-uww-ị heemattʃ-uwwa. ku=baar-ị geeddʒa. ku=beet-ị k’oora. ʔítt’ị heemattʃa. k’ot’arị land-íttʃo bat’ọ guul-i-t-o-ʔ-o. ku=beet-ị k’awwa. tu=land-íttʃ-o k’eeraaʔla-tte. ʔítt’ị dan-aamo. ʔísị dan-aamo-tte. gootoʔi beet-ị min-iínse fir-u-kk-o. gootoʔ-ị min-iínse fir-u-kk-o. k’eeraaʔlị man-tʃ-ị ni baarạ bitaaʔ-u-kk-o. k’eeraaʔl-ị ni baarạ bitaa ʔ-u-kk-o. ʔabbeeb-ị mah-ína t’eeʔ-aallị saatạ bitaaʔ-u-kk-o. saati bít-ị t’eeʔ-u-kk-o. ʔanị hemmattʃị kaamaʔẹ btaaʔ-u-mm-o. ʔad-ị mah-ína heemattʃ-u-kk-o. ʔissuwwị haraarị googọ bat’-am-u-kk-o. goog-ị haraar-u-kk-o. biíʔli ʔuullá-nne heeʔ-aá heeʔ-aá kába. ʔit-aá ʔit-aá geedʒ-ú-kk-o. dʒabb-aá dʒabb-aá leh-ú-kk-o. batt’-aá batt’-aá hoog-ú-kk-o. menk’-aá meenk’-aá neésẹ tʃ’een-siis-ú-kk-o. jak-aá jak-aá ʔítt’i beéti foóreẹ. ka=k’araáreẹ likitʃ’-í-tt-aá-re le-t-oó-jjo. keene mar-ú-mm-í-daʔne ʔub-ú-mm-í-heeʔ-ú-mm-ó-jjo. ʔi-ína ʔammáni heeʔ-u-dáʔne keésẹ. godd-aán-tʃ-ị ʔih-ú-kki-beeʔ-aa-ré-mị bút’ạ. him-ú-kk-aa-ré-mị ʔítt’i míne mar-eéna. ʔánị lobakátị karaáʔlị míne mar-aá-mm-o ʔih-óna bagaánị. biʔ-ú-kkị hall-íttʃọ ʔarák’a has-ú-kk-o. ʔánị gereé-ttʃọ dur-ú-mm-o. ʔi beʃ-íttʃ-ị waar-oó-kk-o. ʔítt’ị geer-oó-kk-o. ka=ʔeeb-akk’-ám-tʃạ lánd-ị sab-am-oó-kk-o. gereé-ttʃọ ʔánị dur-ú-mm-o. ʔísị tʃ’iil-íttʃ-ína wóʔọ ʔuww-i-t-ó-ʔ-o ʔádọ. ʔísị wóʔọ tʃ’iil-íttʃ-ína ʔuww-i-t-ó-ʔ-o ʔádọ. ʔánị ʔitt’é-na moókk’ạ sar-ú-mm-o. ʔánị moókk’ạ ʔitt’é-na sar-ú-mm-o. ʔitt’é-na ʔánị moókk’ạ sar-ú-mm-o. ʔitt’e-na moókk’ạ ʔánị sar-ú-mm-o. ʔátị ʔisé-na diinátẹ ʔúww-e. ʔisé-na hooʃʃóʔọ meent-íttʃọ ʔuww-i-t-ó-ʔ-o. kojj-ína búnạ ʔánị kaas-ú-mm-o. ʔánị beebálla waar-ú-mm-o. beebálla ʔánị waar-ú-mm-o. ʔoo=tte ʔi meent-íttʃó-tte. ʔi land-íttʃọ ʔisé-tte. ʔájj-ị keésẹ haraʔm-ú-kk-o. lánd-ị hínka ʔeeb-akk’-ám-tʃạ doʔl-am-oó-kk-o. ku=mán-tʃ-ị máhạ ʔijj-aá-kk-o. miʃaám-ị ʔajjé-na kitaábạ ʔuww-ú-kk-o. ʔítt’ị hiiʔmóʔo hannó-nne gar-ú-kk-o. ʔi beʃ-íttʃ-ị ʔeésẹ haraʔm-ú-kk-o. lánd-ị heer-ám-tʃạ doʔl-am-oó-kk-o. ku=mán-tʃ-ị tʃ’aná-nne t’aafé?ẹ ʔijj-aá-kk-o. miʃaám-ị makeeb-ína kitaábạ ʔuww-ú-kk-o. ʔítt’ị hiiʔmóʔo ni miné-nne gar-ú-kk-o. beét-ị waar-u-kk-aá-re ʔánị mar-oó-mm-o. wiʃ-íttʃ-ị moóttʃạ siid-oó ʔammáne lobakáta. gaʔn-u-kk-aá-re waar-oo-mm-ó-jjo. ʔeekkide-é kide-é dabaʔl-am-aá ʔarad-am-oó-kk-o. minaadáb-ị beeballa-á kaballa-á kejjé-nne dumm-ú-kk-o. ʔoo=k’eeraal-i-í danaam-i-í makeébi ʔabbaájjo. ʔoo=mín-ị geeddʒa-á haraára. ʔítt’ị ka=beéti lokk-o-ó ta=land-íttʃi horoor-e-é. makeeb-i-í maaddaam-i-í waar-am-oó-lla. ʔoo=geer-am-oo-kk-ó-kkị makkeebo-tte-é liiransó-tte. makkeéb-ị ʔítt’i ʔabbaajj-o-ó ʔíttt’i ʔaajj-a-á. ʔani-í ʔitt’i-í ʔabbaajj-úwwa. ʔíssị ʔeese-é keese-é sog-akk-ó-ʔ-o. ʔítt’ị ka=beet-ína kitaábạ ʔuww-ú-kk-o. ʔánị man-tʃ-o-ó meent-íttʃ-o-ó moo?-ú-mm-o. ʔánị mán-tʃọ moo?-ú-mm-o. ʔi beet-i-í ʔi land-íttʃo-ó biíʔli ʔullá-nne. ʔi beét-ị biíʔli ʔuullá-nne heeʔ-oó-kk-o. ʔítt’ị ka=beet-ina-á ta=land-íttʃ-o-na-á kitaábạ.

Appendix B: The Sentence Extraction Algorithm

To extract the first sentence from a file

1. Initialize a variable, input string to hold the entire content of the file.
2. Initialize a variable, SentenceCounter, to hold location.
3. Assign the entire content of the file to the variable initialized, i.e. inputstring, using the built in string manipulation function.
4. Starting from the first word in the inputstring, extract each word up to the end including the full stop, question mark or exclamation mark (., ? or !) from the inputString.
5. Assign what you extract in step 4 to a variable, inputsentence, which holds only the sentences extracted. (Note that at this stage one sentence is extracted.)
6. Update the SentenceCounter by 1
7. Assign the sentence extracted in step 5 to a text box, unparsed sentence text output box, to display the first sentence extracted
8. Read the previous sentence.
9. Repeat step 4 onwards until the end-of-file mark is found.

Appendix C: The word Extraction Algorithm

When parse button is pressed from GUI, a function that uses the built in string manipulation is used to extract the first word from the input text. Then the following procedures will be executed to split all words of the sentence

1. Initialize a variable to hold the individual word
2. remove all spaces between words in a sentence
3. remove also ENTER
4. if the sentences do not end with a full stop “.” Or a question mark “?” or exclamation mark “!”
5. assign POS tag for the word
6. check whether the sentence starts with after end marker punctuation
   i. if the sentence starts after end of sentence marker
   ii. set Top with the letter “S”
   iii. end if
7. split the sentence into words
8. else call the parsing function

Appendix C: The Chart Algorithm

When the Parse button is pressed, the Chart-based algorithm is executed in three steps, Arc introduction, arc addition and chart parsing steps. This algorithm is a modified version of (Allen, 1995) algorithm for chart parsing.

The Arc introduction Step

The initialization step will use the following steps for each word identified using the sentence Splitting algorithm discussed above.

1. initialize sentence position, word counter and head rules probability
2. start reading the sentence from the input
3. fetch the grammar rule from the database (text document)
4. if the grammar rule in the database (text document) is not empty
5. for each sentence, get information about sentence head rules from database (text document)
6. for each word
 get word category information from database (POS model)

if there is no word information in the database (POS model)
analyze the new word into its root and morpheme form
check the database (check parse model dictionary) for stem information
 else get word category from the database
 end if
assign grammar rule of a given word category from the database(Text) if it is not empty
collect every constituent of the grammar rule and add them to temporary list each
make it active arc (* incomplete arc i.e. that looks for addition of a constituent to be complete its arc
select the completed arc and add them to the main arc
if the main arc is completed, read the next word for its interpretation
 end if
 5. Repeat step 3 on wards until the end of the sentence.

The constituent Addition Step

The Constituent addition step follows the following procedures.

1. Initialize a variable, ActiveArc
2. Starting from the next word, i.e. next to the word identified in the initialization step, to the last word the following steps will be executed iteratively (extracted from (Allen, 1995))

3. To add a constituent C from a position P1 to P2:

1. Insert C into a chart from position p1 to P2
2. for any active arc of the form X→ x1, ….c…xn from position P0 to P1, then add a new arc X→ x1, …. C …xn from position P0 to P2
3. for any active arc of the form X→x1, …xn …. c from position P0 to P1, then add a new constituent of type X from P0 to P2 to the agenda.
4. do until there is no input left:
   if the agenda is empty, look up the interpretations for the next word in the input and add them to the agenda select a constituent from the agenda (let's call it constituent C from position P1 to P2)
   for each rule in the grammar of form X → C x1 … xn, add an active
   arc of form X →C x1 … xn from position P1 to P2.
   add C to the chart using the arc extension algorithm above
4. finally clear the grammar rule from the temporary Arc
5. do this iteratively until the end of grammar rule and end of words in the sentence
6. else if parsing has reached the end position then write the end position

7. start selecting the right arcs and add them to the variable called, I array
8. call the tree construction function
9. while constructing tree check for agreement by calling the analyze agreement function
10. display the output of this process

The Steps in Chart parsing

The chart parsing step follows the following steps

1. get the sentence rule from grammar rule table and word category from Lexicon model
2. Assign the value of rule in the data source to sentence rule (head rule)
3. if the active arcs are not found
4. exit
5. otherwise, split the grammar rules into its components
6. for each main arcs component and for each active arc assign the main arc component to the left hand side grammar rule
7. check whether the parse is correct or not
8. if right load the active arc introduced and read the word with root form to get word information from the Lexicon table
9. if active arc is found the parse is correct
10. else report, not found message
11. Starting from the next category to the last category it will execute the following steps iteratively

3.1 Assign the value in the raw of the current category column last index to a variable max1.
3.2 if the value of max1 is greater than the value of max compared in step 1 of this phase above, then change the value of max with max1 and assign the index of the category to the variable loc.
3.3 if there is another category start again from step 3 of this phase
3.4 in another array called C it does the following procedures
3.4.1 assign the loc. value of lastword +1 found above at the place of last index in the array C
3.4.2 for each word string from lastword-1 up to the first, assign the corresponding location value from BACKPTR array to a corresponding place in array C

Appendix E: Morphological Analyzer Algorithm

The morphological analyzer does the following things to help the parser during parsing.

1. initialize the variable morph
2. get word category from the function called word category
3. read the first word from the sentence
4. if the word is at sentence position 1 then the word is noun, or adjective
   but if at sentence position 2 then the word is a verb
5. assign the value of full word from the sentence to stem
6. if the string for example '-nne' is appended detach it first
7. desuffix function for'-nne'
8. check for number of the word using word information and Affix list table(in a dictionary)
9. Do for other features(POS tags like, NN(Noun), VB(Verb)) the same iteratively until the end of the sentence
10. finally print the result to the parsing function

Appendix F: The Tree construction and Extraction Algorithm

Finally, the Parse-text function will execute to produce words with their appropriate Parsing as out put

1. split the right hand side rule of the main arc
2. fill the first level children in the tree
3. fill the rest deep components
4. do
    a. collect the deep arcs in the tree
    b. expand the right hand side rule
    c. load the children to the right hand side rule node
    d. repeat until the terminal node is reached

Tree extraction step

Fetch the right hand side rule from the database (text document)

Split the right hand side rule and assign index to each

Extract tree based on the index value of the rule

Now analyze the agreement case (for number, gender, case, tense and display the output in the tree form and grid form.

Appendix G: Lists of Part-of-Speech tags

List of part-of-speech tags used in the research Automatic Supervised Sentence Parser for Hadiyya Language:

Number	Tag	Description	Example
1.	CC	Coordinating conjunction	teʔíme
2.	CD	Cardinal number	1, lamo
3.	CCP	Coordinating conjunction pronoun	beetína
4.	DT	Determiner	Ku, kúkị
5.	FW	Foreign word	baabura
6.	IN	Preposition or subordinating conjunction	Bakko, laso, taa, bagaánị
7.	JJ	Adjective	hemmattʃị, mett'oʔo
8.	JJR	Adjective, comparative	hemmattʃa
9.	JJS	Adjective, superlative	hemmattʃukkoki
10.	LS	List item marker	1)
11.	MDVB	Modal Varb	waaroókko
12.	NN	Noun, singular or mass	meera,
13.	NNS	Noun, plural	Laro, goddaáno
14.	NNP	Proper noun, singular	Shaamebo
15.	NNPN	Proper noun Negative	taaddasattéjjo
16.	NNPS	Proper noun, plural	laro
17.	PDT	Predeterminer	lami gooni
18.	POS	Possessive ending	ʔítt'ị
19.	PRP	Personal pronoun	ʔate, ʔane
20.	PRP$	Possessive pronoun	ʔanị, ʔi
21.	RB	Adverb	Sooddanna, hundiʔammáne, daridára
22.	RBR	Adverb, comparative	mataage
23.	RBS	Adverb, superlative	ʔarák'a
24.	SYM	Symbol	$, %
25.	VB	Verb, base form	mass
26.	VBD	Verb, past tense	mass-akk-o, mar-ú-kk-o
27.	VBDN	Verb, past tense negative form	waar-am-ú-kk-jjo
28.	VBG	Verb, gerund or present participle	mass-i-t-aa-mm-ulla
29.	VBN	Verb, past participle	mass-akk'-u-kk-o
30.	VBP	Verb, non-3rd person singular present	mass-e
31.	VBZ	Verb, 3rd person singular present	mass-ehe
32.	WDT	Wh-determiner	hínkị, hínka
33.	WP	Wh-pronoun	ʔájje, máhị
34.	WN	Wh-noun	taddasattenníhe
35.	WP$	Possessive wh-pronoun	ʔájji
36.	WRB	Wh-adverb	hinkeʔe, hinki-ʔammáne

Appendix H: Head Rules for SASPFHL Hadiyya sentence parsing system

S-> NP PRP
NP NN ADVP RB NP NN NN VP VBD
ADVP RB NP NN NN VP VB
NP WDT NN NN VP VBD
NP NNP NN NN
NP FW
NP WP NN NN
NP NNP NNP VP VBD
NP NNP WDT VP VBD
NP NNP NN NN VP VBD
NP NNP NN WDT VP VBD
NP NNP NN VP VBD
NP NNP ADVP WRB
NP NNP VP VBD
NP NNP NN ADJP JJ
NP NN WDT ADJP JJ
NP DT NNP
NP PRP NN
NP NNP ADJP JJ
NP NNP$ ADVP WRB
NP DT NN ADVP RB
NP NN PRP WP$
NP PRP ADVP RB
NP PRP NN NNP
NP PRP NNP$
NP DT NN NN
NP DT PRP NN
NP DT NN WDT
NP PRP NN NN ADVP RB
NP PRP$ NN ADVP WRB
NP PRP NN WDT
NP PRP NN ADVP WRB
NP PRP
NP PRP WN
NP DT PRP VP DT VB
NP DT PRP WN
NP DT WP
NP DT NN WP
NP PRP WP
ADVP RB
ADJP JJ NN
NP DT PRP NN VP VBD
NP PRP NN VP VBD
NP NNP$ NN VP VBD
NP NNP ADJP JJ NP NN
NP DT PRP ADJP JJ
NP DT NN ADJP JJ

NP DT NNS JJ NNS
NP PRP NNS
ADJP JJ NP NN PRP NN VP VB NP NN VP VBD
NP PRP VP VB NP NNP VP VBD
NP PRP VP VB VBD
NP PRP VP VB NP NN NN PRP VP VBD
NP PRP VP VB NP NN NN VP VBD
VP VB ADJP JJ NP NN ADVP RB
NP DT NN DT PRP VP VB NP NN VP VB
NP DT PRP DT NNS DT NN VP VB
ADVP RB VPP VB VB
NP NN NNP NN PP IN IN
NP NNP ADJP JJ NP PRP NNP$ NN VP VBD
NN NN VB VBD PRP NNS VB VBD
ADJP JJ JJ NP NN PRP$ VP VB NN ADVP RB ADJP JJ NP NN VP VBD VBN
NP NNS PRP ADJP JJ NP NN VP VBD
NP NNP ADJP JJ NP PRP NN VP VBD
NP PRP NN VP VB NP NN VP VBD
NP NN VP VBD NP NN VP VB VBD
NP PRP$ NN NN VP VBD ADJP JJ NP NN VP VBD
NP NN VP VBG VBG
NP PRP$ NN NN VP VBZ ADJP JJ NP NN VP VBD
ADVP RB NP NN VP VB VP VB
NP NNP NN VP VB VB ADVP RB RB VP VBD
VP VBD VBD VBD
ADJP JJ JJ VP VBD
VP VB VB VBN
ADVP RB RB NP PRP PRP NN VP VBD
NP DT NN VP VB VBD
ADVP WRB VP VBZ MDVB
NP POS ADVP WRB RB NP PRP VP VBN
VP VB ADVP RB VP VB VBD
ADJP JJ NP PRP NN VP MDVB VBD
PRP JJ NN NN VBD CC IN VBG
ADVP RB NP NN ADVP RBS VP VBD CC IN VP VB VBD
NP PRP NNP VP VBD
NP PRP$ NN VP MDVB
NP PRP VP VBD
NP DT NN PRP VP VBD
NP NNP PRP VP VBD
NP PRP NN NN VP VBD NP NN VP VBD
NP PRP PRP NN VP VBD
NP PRP NN PRP VP VBD
NP PRP PRP NN VP VBP
NP PRP ADJP JJ NP NNP VP VBD NP PRP VP VBD
NP NNS NN PRP VP VBD
NP NN PRP VP VBD
NP WDT PRP$ NN

NP PRP$ NN PRP
NP WP PRP VP VBD
NP PRP WDT NN VP VB
NP DT PRP WDT VP VBN
NP NNP WP NN VP VBD
NP PRP NN WDT VP VBD
NP PRP$ NN PRP VP VBD
NP PRP ADJP JJ VP VBD
NP DT PRP ADJP JJ NP NNP VP VBN
NP NNP PRP NN VP VBD
NP PRP NN PRP$ NN VP VBD
NP NN VP VBG NP PRP VP VBG
NP NN NN ADVP RB RB RB VP VBD
VP VBN VBD
ADVP RB RB VB VBD
NP NNP NN NN VP DB VBD
ADJP DT JJ JJ NP NNP NN
NP DT NN ADJP JJ JJ
NP PRP DT PRP NN DT PRP$ NN VBD
NP NNP NNP NP VBG
VP DT VBG NP NNP NNP
NP NNP PRP NNP PRP VB VBD
NP PRP$ PRP$ NNS
ADVP RB RB RB NP NN
NP PRP DT NNP$ NN VBD
NP PRP NN NN VBD
NP PRP NN VBD
NP PRP$ PRP$ PRP NNP NN VBD
NP PRP$ NN NNP NN VBD
NP PRP DT PRP DT NN NN VBD