JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

EVENT AND TEMPORAL INFORMATION EXTRACTION FROM AMHARIC TEXT

By: Ephrem Tadesse

THESIS SUBMITTED TO FACULTY OF COMPUTING JIMMA INSTITUTE OF TECHNOLOGY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION TECHNOLOGY

September, 2017

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

EVENT AND TEMPORAL INFORMATION EXTRACTION FROM AMHARIC TEXT

BY: Ephrem Tadesse

ADVISOR:

Mr. Debela Tesfaye (PhD candidate)

CO-ADVISOR:

Mr. Tesfu Mekonen (MSc)

APPROVED BY

1. **Mr. Debela Tesfaye, Advisor**
2. **Mr. Tesfu Mekonen , Co-Advisor** ___________

Approval sheet

This independent research entitled as “Event and Temporal Information Extraction from Amharic Text” has been read and approved as meeting the preliminary research requirement of school of computing in partial fulfillment for the award of the degree of masters in Information Technology.

Jimma University, Jimma, Ethiopia

Advisor	Signature	Date
Debela Tesfaye(PhD candidate)		11/13/2017
Co-Advisor		
Tesfu Mekonen(MSc)		11/13/2017
External Examiner		
Dr.Dereje Teferi (PhD)		11/13/2017
Internal Examiner		
Getachew Mamo (PhD candidate)		11/13/2017

Dedication

For mom: Mama ,you are my everything, I have no words to express your love, patience and motive You made me feel like me again and always I wish to be with you to have a real joy. Wish you long live mama ewedshalehu.

For Dad: Ababi, you are special for me, you are not only my father you are my big brother, you are my friend, and you are my light. You showed me what a wonderful father mean for his family; I feel proud being your child. Wish you long live ababi ewedhalehu.

Acknowledgement

Foremost, all the praise is going to the almighty GOD who strengthens me to pass every challenge in my life smoothly and to finish this work successfully. It is with immense gratitude that I acknowledge the support and help of my advisor Mr. Debela Tesfaye for his motive and invaluable comment to do this work from the beginning to end. I would also like to thank my co-advisor Mr. Tesfu Mekonen for his constructive comment and his courage empowers me to do this work. I extend my gratitude to Mr. Daniel Yacob for his motive and sharing of valuable ideas. I would also like to thank everybody who encourages and helps me in this work especially my friend Mizanu Zelalem, my classmates , my colleagues, and all other guys whom I didn't list their name thank you for your motive and tremendous help. Last but, not the least, I would like to thank my family: mama and ababi you are my life. It's impossible to be there without your incredible help.

Table of Contents

List of Table

List of Figure

Acronyms & Abbreviations

ETIEA: Event and Temporal Information Extraction from Amharic Text

IR: Information Retrieval

IE: Information Extraction

MUC: Message Understanding Conference

NER: Named Entity Recognition

ACE: Automatic Content Extraction

TE: Template Production

TR: Template Relation Construction

TE: Template Element Construction

GATE: General Architecture for Text Engineering

SRL: semantic role labeling

SERA: System for Ethiopic Representation in ASCII

RDF: Resource Description Framework

RDS: Amharic Rule Discovery System

POS: Part of Speech Tagger

LIBSVM: Library of Support Vector machine

kNN: K-nearest Neighbor

DT: Decision Tree

NB: Naive Bayes

ROC: Receiver Operating Characteristics

WIC: Walta Information Center

CVCVC: consonant vowel consonant vowel consonant

Abstract

The drastic increase of large volume of data on the web becomes cumbersome to get relevant information. To tackle this problem a lot of information extraction tasks have been done from the literature background. Event and Temporal information extraction is one of information extraction tasks, which helps to get important events from large set of texts with their chronological order and answers the question of what happened on a certain situation as well as when does it happen. Unlike other information tasks like entity extraction research needs felt for event and temporal information extraction especially for Amharic still there is no work on this particular IE task.

As the first comprehensive work we designed a model on event and temporal information extraction from Amharic text. The model is comprised of different components including common preprocessing, learning and classification, event extraction, temporal information extraction. To develop the proposed model we used different approaches for each tasks. For event extraction component we used a machine learning classifier but the classifier fails to detect deverbal events. To resolve the machine learning classifier limitation of missing deverbal entities due to their ambiguities we used rule based approach using syntactic features such as POS, morphological analyzer, and list of gazetteers. In practice it's difficult to stay within the boundary of single event extraction method. So as both approaches have advantages and disadvantages combining those results to get the advantage of the machine learning classifier and the rule based approach advantage we developed hybrid approach for event extraction. For temporal information extraction component regular expression, list of temporal gazetteers in combination with some rules is used. The preprocessing component is used to prepare and normalize input texts. Whereas the event extraction component extracts events and the temporal information extractor is used to extract and normalize temporal expressions. Various experiments are conducted for each approach with different scenarios. The hybrid approach for event extraction component outperforms over the other two approaches and the evaluation result yields precision, recall, and F-measures of 97.7%, 96.3% and 96.99% respectively. The rule based approach for temporal information extraction scores a precision, recall, and F-measures of 84.6%, 89.7% and 87.1% respectively.

Key words: Event and Temporal information Extraction from Amharic Text, Information extraction, machine learning classifier for event Extraction, Rule based approach for Event Extraction, Hybrid approach for event extraction, Rule based temporal information extraction, Deverbal entities, Regular expression.

CHAPTER ONE

1. INTRODUCTION

Humans developed natural language to communicate; over past millennia, it has been the most efficient form of transferring the majority of information between individuals [7]. Natural language processing (NLP) is a sub field of Artificial Intelligence which has the ability of computer systems to analyze and synthesize spoken and written languages as human beings [16]. The word ‘Natural’ is used to distinguish those formal languages like programming languages from that of the task performed by NLP. Amharic is a Semitic language which has around 27 million speakers as a mother tongue and others with different mother tongues in different parts of Ethiopia [10]. It is also the official working language of the federal democratic republic of Ethiopia. NLP has a wide range of applications that make use of NLP to allow users interact with computer systems using natural languages. The most important and widely used applications are information retrieval, information extraction, question answering, machine translation etc.

Nowadays, large volumes of data are available on the web. But, getting required facts from these sources become a challenge; therefore, Information Extraction was developed to extract pre-specified information from the raw text in order to organize it in structured representations such as templates or databases [1]. The major tasks in information extraction are determining the information unit to be extracted and establish the extraction patterns with the help of human intervention. The Message understanding conference (MUC) have laid out tasks which are considered as generic tasks that allow the evaluation of different IE systems [10] , so that those systems are evaluated based on the performance of the tasks set. The dominant tasks listed are Named Entity Recognition, Temporal and Event Processing, Relation detection and extraction, Template filling. Our work mainly focuses on temporal and Event processing.

Many of the natural language analysis systems focus on nouns and noun phrases in order to identify information on who, what, and where[53]. Moreover other than MUC (1992) most of the recent information extraction conferences and historical frameworks mainly concentrate on named entity task, which identifies nouns and noun phrases. Although rich information on nominal participants, actors, and other entities is provided, the named entity task provides no information on what happened in the document, i.e. the event or action. Less progress has been made on ways to utilize eventual information.

Temporal location and ordering of events are mostly expressed in newspaper texts and narrative style documents. Text comprehension requires the capability to identify the events described in a text and to locate them in time. So, to address temporal structure of a text identification of temporal expressions and identification of events has paramount significance. In [8] Event extraction is a popular and interesting research field in the area of Natural Language Processing. It is one of the most important and difficult tasks in IE which identifies events and related information from the raw text. Event extraction from news domain helps to enhance the performance of personalized news systems, in such a way news would be selected accurately based on the preference of users and selected events [2]. The basic concepts behind event Extraction are Events, Event Trigger, Event Argument and Event Mention.

The Oxford English Dictionary defines an event as *"*a situation that takes place or lasts for a certain period of time, especially one of importance". Independent of their form of expression, events may be taxonomised into discrete classes such as: occasions, actions, occurrences, reports, perceptions, states, intentional actions, aspectual. However, the occasions, actions, occurrences and states are used in natural language more widely than this definition permits; there are often mentions of negated events, conditional events or modal events, which cannot be said to certainly "happen or take place". Events in natural language text represent happenings and actions that occur at a specific time and place. Generally, events are described in different newspaper texts, stories and other important documents where events happen in time and ordering of these events are specified.

Temporal information in text is often expressed using a phrase that precisely describes a point or duration. Sometimes these points reference an absolute unambiguous time (anchored via e.g. a calendar), which is of great help when trying to map events from a discourse to a timeline [7]. According to information theory science timeliness or currency is one of the key five aspects that determine a document's credibility besides relevance, accuracy, objectivity and coverage. Temporal information extraction plays a crucial role in improved information access, in particular for creating timelines and detailed question answering. There are different types of time denoting expressions in a document such as: Explicit reference (date expressions), indexical reference: all temporal expressions that can only be evaluated via a given index times are said to be indexical.

There are works that has been done under different languages for automatic extraction of events and temporal expressions from unstructured text. But, we cannot directly apply the works done for other language for Amharic due to linguistic features variation and domain specific issues. This initiates us to develop framework for automatic extraction of events and temporal information from Amharic unstructured text. Our work is the first time in this particular IE application for Amharic language. We try to compare the state-of-art approaches for other languages from the literature background. Our approach is based on the linguistic criteria and overview of other literatures result in this particular domain. The output of our work will be extended for other NLP applications like question answering, text summarization, and risk analysis etc.

1.1. Statement of the Problem

Amharic is one of the Semitic languages which have large numbers of speakers next to Arabic and more than Hebrew around 27 million speakers as a mother tongue and others as second language [11]. Currently, it serves as the official working language of the FDRE. The Ethiopian government has made the development of ICT one of its strategic priorities the amount of documents which are stored electronically has shown drastic progress and the use of computers to do everyday tasks. Most of these electronic documents contain lot of important information in unstructured format. Unlike Amharic for other languages various approaches are used in the literature to automate the process of event extraction so as to get events in chronological order.

Event and temporal information extraction is important tasks of information extraction in which events describes structured templates that relate an action to its participants, times, places and other entities. The time expressions are very important to map events with the discourse timeline based the document type. So, extracting events with temporal expression in their national and local relevance is helpful for newswire domain. The output of this work will be extended in many different NLP applications like Question answering, Text entailment, Automatic text summarization, Decision support system and Risk analysis etc. Event and temporal information extraction has been done several times for few languages. But, still now there is no any reference for Amharic, so our work is the first time to develop automatic Event and Temporal expression extraction. The research question will be addressed in this work are:

- What are the appropriate features that helps the machine learning classifier to predict correct class of instances
- What seems to be the model suitable for automatic Event and Temporal information extraction from Amharic text

1.2. Objective

1.2.1. General Objectives

The General objective of this thesis work is to develop automatic event and temporal information extraction system from Amharic text.

1.2.2. Specific Objectives

The specific objectives which enable us to fulfill the general objectives of our work are listed as follows.

- Reviewing various literatures that has been done under different languages for Event and temporal information extraction
- Review and analysis of Events and temporal information in Amharic language texts
- Data collection, preparation, normalization and conversion of Amharic language characters , time expressions
- Event detection as classification of sentences
- Feature extraction for machine learning classifier
- Disambiguate deverbal entities
- Ethiopic Number Conversion to Arabic number
- Normalize temporal expressions
- Develop complex regular expression for temporal information extraction
- To design and develop a model for event and temporal information extraction from Amharic text.

1.3. Research Methodology

Generally our research method can be considered as being part of a quantitative method, in which the answer to the research question can be quantified respond through constructing models. But specifically our research belongs to the sub category of quantitative research method, i.e. experimental quantitative research method. In experimental research method the answer to research questions is obtained by conducting various experiments. We conduct different experiments to explore the performances of automatic event and temporal information extraction models built using our framework.

1.3.1. Literature Review

Various works of literature and related works that have been done in the area of information extraction, related NLP applications and tools under event and temporal information extraction are reviewed to understand the state-of-the-art approaches.

1.3.2. Data collection

Any kind of research conducted in NLP requires resources and among those resources data is prominent. Since most of the state-of-art researches are conducted using data driven approach which are highly depends on large amount of corpus, it is obvious why data is crucial. But, for Amharic, which is a resource poor language, shortage of standard corpus is a major problem encountered when conducting NLP research. Although, there is an ongoing research on building morpho syntactically annotating Amharic document to build Amharic Treebank in the work of [18], we couldn’t find their corpora due to the unfinished work and willingness of the authors to feed their progress work for the research work. Rather our data source is Amharic corpus which is prepared by the Ethiopian Languages Research Center of Addis Ababa University in a project called "The Annotation of Amharic News Documents”.

1.3.3. Data Preprocessing

Data preprocessing is a preliminary task in any text processing applications. It’s not trivial task especially for languages which are morphologically rich and poses complicated syntactic features like Amharic. At preprocessing step data is converted to appropriate format required for IE process. Since, we deal on unstructured text there are irrelevant information in a document that will cause loss of performance for the system and memory consumption. To get ride off that irrelevant information preprocessing tasks including Stop word removal, tokenization, sentence splitting, character normalization, Ethiopic number conversion etc are performed.

1.4. Modeling tools and Techniques

Debating is continued in IE environment on the superiority of data-driven and knowledge driven approaches for event extraction. Although for both approaches relative performances have been reported in literature in terms of precision, recall, and *F* measures, advocates of data-driven techniques emphasize their favorable computability, whereas knowledge-driven approaches are encouraging a higher degree of interpretability due to the general traceability of the results. Users of hybrid event extraction approaches, on the other hand, effectively combine both approaches to gain their advantage. In this study based on different considerations such as language characteristics, time constraints, dynamicity of the method, the application we are going to do and suitability of the method for our work we decided to use both the favors of data-driven and knowledge oriented approaches. In order to develop a prototype system, different appropriate tools are used. The Java programming language is used to implement the different language specific algorithms developed.

1.4.1. Supervised Machine learning classifiers

We employed a supervised machine learning classier to extract events. Events are detected as classification. We have binary classifiers to classify terms as one-event or off-event. Different feature sets are applied for the classifier which enables to classify as either of the two predefined classes.

1.4.2. Argument Extraction

Argument extraction is a complex and challenging task in an event extraction. There are different arguments like place, time, agent and event. For our case we mainly focus on time arguments.

1.4.3. POS Tagger

Since there is no any readymade syntactical parser for Amharic instead we used pos tagger which helps us to analyze syntactic features of words with their respective word class. Based on the classes of words that is generated as model considering their context in different instance. Now we can be able to extract event trigger keywords in a given text by incorporating language specific rules.

1.4.4. Morphological Analyzer

Morphological analyzer is used to analyze words into their principal morphemes and it generates tokens, given stem and a picture of the word's grammatical structure. Some of the NLP applications including information retrieval, question-answering, speech recognition, and machine translation systems, rely on a lexicon of possible forms in certain language. Analyzing morphology is crucial for morphologically rich languages as such Amharic. Events in a sentence are described in many ways mainly as scholars argue event trigger keywords belongs to verbs and nouns sometimes adjectives. Nominal events are ambiguous to extract because of their classes and context. To disambiguate those deverbal and non deverbal nominal we adopt Morphological analyzer which is an open source python project done for three languages Amharic, Afaan Oromo and Tigrgna [67].

1.4.5. Temporal expression extraction

There are different cross-domain sensitive temporal tagger tools publically available. The popular ones are SUTime and Heidel Time Tagger. Heidel time tagger owns the full task of temporal tagging which implies accurate extraction and standardization of temporal terminologies. So that most of the researchers choose Heidel time tagger to achieve the best result in their work. The selection criteria is based on the domain dependency strategy in which, Heidel Time uses domain-dependent strategies and achieves high-quality normalization results on both corpora. This domain sensitive tool works on a complex rules and regular expressions in addition to dictionary of temporal trigger keywords. Since, temporal information in a text is not huge unlike that of the other linguistic features. As such we develop our own language specific rules to extract temporal information from Amharic unstructured text. Based on the literature background rules are efficient to extract the temporal expressions in a text. In this work, we employ robust regular expression, rule-based approach and we incorporate temporal trigger keyword gazetteer to get the frequently occurred temporal keywords.

1.5. Analysis and Evaluation

Analysis of the results and evaluation of the performance is crucial and the target for the conclusion in any system. The popular evaluation metrics used in the information extraction environment are precision, recall, F-measure and sometimes ROC (receiver operating characteristics) [3]. The output and performance of any IE application relays on the quality of the input data and the techniques used. Since event and temporal information extraction is one of IE application for our work we need to apply the same metrics.

1.6. Scope and Limitation of the study

Event and temporal information is one task in information extraction. Unlike entity extraction, it didn't get too much focus from the literature background. In order to develop a consistent model for this task, we need to have other readymade interdependent NLP components such as sentence parser, coreference resolution, and NER etc. Even though some of the NLP components for Amharic language have been done by some researchers, they are not publicly available and readymade to employ them as extension confidentially. Assuming the overall constraints the scope of our work is developing a system for event and temporal information extraction from Amharic unstructured text. Extraction of the relationship between events, the relationship between event and time, ordering of events, the relation between document creation time and event are out of the scope of our work due to the above constraints. Unstructured texts are considered as a target for this work.

1.7. Application of Results

Event and temporal information extraction from unstructured text such as news messages could be beneficial for IE systems in various ways [2]. For instance, being able to determine events could enhance the performance of personalized news system. The following are application areas of this study:

- Question answering: the output of our work will be extended to answer when and what WH questions by developing query based dialogue management for the user
- Textual entailment:
- Medical Domains: biological events are extracted and this domain is beneficiary currently there are works that are going on to extract biological events from clinical data. Diagnostic and treatment could be indeed enhanced by reviewing patient history synthetically in the order in which medical events occurred [61]

- Automatic text summarization: enables to summarize a text easily based on the most important events and participants.
- Decision support system: informed decision support system
- Risk analysis: analysis severity of risks according event types
- Algorithm trading: in such a way representing the use of computer programs for entering trade orders with algorithms deciding aspects like timing, price, and quantity of an order.
- Economic events like mergers and acquisitions stock splits dividend announcements
- Financial markets are extremely sensitive to breaking news for informed decisions

1.8. Structure of the Thesis

In summary this work is organized as follows. Chapter one presented introduction to the research background of ETIEA, problem statement, specific and general objectives, research methodology, scope and limitation, and finally application areas of the result. Chapter two talks all about the theoretical background of an IE system as Literature review including IE tasks, components, dominantly existing approaches with their category and the necessary evaluation metrics for IE system. Chapter three is about related works done in event and temporal information extraction using different approaches under different languages. Chapter four discuss the Amharic language structure including common linguistic characteristics such as punctuation marks, major word classes, Amharic verb morphology, events and temporal information extraction in amahric text, some normalization schemes etc. chapter five contains all about the main components of this work. It is comprise of design of the prototype and implementation issues along with their functional operation of components and subcomponents. The approaches used and the algorithms developed are briefly described in this chapter. Chapter six contains the experimentation and evaluation of this work. Finally chapter seven discusses the contribution of this work, future works and conclusions.

CHAPTER TWO

2. LITERATURE REVIEW

In order to understand the problem domain from the literature background and to identify clear boundary of our works from the current state-of-arts different books and research works, which are related to information extraction, event and temporal information extraction and related fields have been reviewed.

2.1. Information Extraction

Information Extraction is a subarea of NLP, which extracts facts from text and puts them into structured representations such as templates or databases. IE research has been promoted by U.S. Government sponsored programs (MUCs and ACE). The growing availability of on-line textual sources and the potential number of applications of knowledge acquisition from textual data has led to an increase in IE research.

IE is a technology which depend on analyzing natural language in order to extract snippets of information. It is the task of finding structured information from unstructured or semi-structured text. The process takes texts (and sometimes speech) as input and produces fixed format, unambiguous data as output. This data may be used directly for display to users, or may be stored in a database or spreadsheet for later analysis. Sometimes it is used for indexing purposes in Information Retrieval (IR) applications such as Internet search engines like Google. IE is quite differing from IR system in which an IR system finds relevant texts and presents them to the user; while, an IE application analyses texts and presents only the specific information from them, which the user is interested in [22]. The information extraction paradigm has much in common with the field of IR and has adapted a lot of standard evaluation metrics from IR including precision, recall, and F-measure.

The historical perspective of IE explained that MUC was sponsored by DARPA in the 1980's that contributed a lot to the popularization of Information Extraction as it is the one that defined the tasks involved in IE and also the evaluation of IE systems using standard evaluation metrics. The MUCs (Message Understanding Conferences) [31, 32] sponsored by the U.S. Government, were the first attempt to standardize the task of Information Extraction and establish benchmark corpora. Starting with the first MUC in 1987, there were seven MUC evaluations carried out in a ten years span. These evaluations greatly promoted the research in information extraction. The goal of the MUC program was to provide a platform on which various IE approaches can be compared. In each evaluation, training data, test data and a scoring metric were provided to participants. Later evaluations were designed to be carried out with a real scenario constraint: for MUC-6 and 7, an IE.

The ACE (Automatic Content Extraction) evaluation program is a successor to the MUCs. After the development of IE in the MUCs, the tasks of information extraction and their difficulties became better understood by researchers. So, the ACE program designed to form basement for each component of information extraction and eventually lead to the extraction of content from text. Therefore the tasks of information extraction have been tackled in a bottom-up fashion. The ACE program started from the building blocks of content extraction, such as named entity recognition and entity co-reference resolution, and is moving up to entity relation detection and event extraction.

2.2. A general Information Extraction System Architecture

Although information extraction systems that are emerged for different tasks often differ from each other in many ways. There are core elements those are shared by every extraction system. Regardless of whether it is designed according to the Knowledge Engineering or automatic training paradigm the common elements are shared. So that every information extraction system has four primary modules, namely a tokenizer, lexical and or morphological processing, syntactic analysis, and some sort of domain-specific module that identifies the information being sought in that particular application. Actually, some extraction systems like name taggers stop at the lexical/ morphological stage, but we are considering systems targeting events and relationships [43].

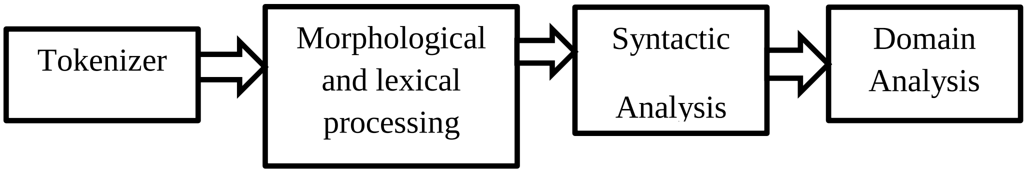


Figure 2-1 General Architecture of IE

2.3. Information Extraction Tasks

The task of IE is to identify instances of a particular pre specified class of entities, relationships and events in natural language texts. In addition to this extraction of the relevant properties of the identified entities, relationships or events are the major tasks of information extraction system. The information to be extracted is pre specified in user defined structures called templates or objects, each consisting of a number of attributes, which are to be instantiated by an IE system as it processes the text. The slots fill are usually: strings from the text, one of a number of pre-defined values, or a reference to a previously generated object template. One way of thinking about an IE system is in terms of database population since an IE system creates a structured representation (e.g. database records) of selected information drawn from the analyzed text [23].

2.3.1. Named entity recognition

Natural language Text usually contains all kinds of names, for example person names, company names, place names, sports teams, chemicals including other names from a specific domain. Other common units can also fall into this category, such as time expressions, numbers or job titles. Names are referred to as named entities in Information Extraction. Failing to recognize them as a unit would affect the accuracy of deeper analysis of text, such as chunking or parsing. Therefore named entity recognition becomes a basic component technology for Information Extraction or Natural Language processing in general [29].

The main tasks done by named entity recognition is to find the span of a name and determine the type of the entity. The type of named entities also depends on the domain and the tasks of information extraction. There are seven types of common NEs defined by MUC for newswire text. Entities include person name, organization name, place or location, date-time expressions, money and percentage are dominant. Various techniques has been employed to recognize named entities among them the data-driven approach is applied to detect NEs in text, including supervised machine learning algorithms such as Decision Tree, Maximum Entropy, and Hidden Markov Model and recently support vector machine were applied to this task with good performance reported [30].

2.3.2. Co-reference resolution

It is a process of finding multiple references to the same object in a text. It refers to the task of identifying noun phrases that refer to the same extra linguistic entity in a text. It is important to note that the same thing about a single entity is expressed in different sentences using pronouns [34].

2.3.3. Event extraction

Event extraction is a sub-problem of Knowledge Extraction which aims to extracts meaningful information called events in the form of situations, occurrences, action, circumstances etc from raw text. Events are also categorized into semantic classes based on their temporal behavior. Extraction of events and its semantic classes requires deeper understanding of syntactic and semantic information of the text. This task includes extraction of an event and all of its arguments. For example, for a terrorist attack event, we expect to extract the time, location, perpetrator target, damage, victims, etc. An event often spans several sentences, so we may need inference to figure out all the slots of an event. Logic and real world knowledge are often needed in this process. Extraction of an event is a challenging task given the variations of human language. Literature presents an abstract view of what Events are and there is no a proper definition to Events. Events are categorized either of under the following classes:

- **Situations:** Real Word occurrences that happen or occur. For example terms like war, floods, launch are events that describe situations
- **Actions:** Process of doing something. For example terms like operation, attempt, mission, assistance, and offer etc. denotes action events.
- **States:** Condition or circumstances that someone or something is in at particular time like believe, kidnapped and sick etc.

2.3.4. Template Relation Construction

Before MUC-7, relations between entities were part of the scenario-specific template outputs of IE system evaluations. In order to capture more widely useful relations, MUC-7 introduced the TR task. As described in [51] "The template relation task requires the identification of a small number of possible relations between the templates elements identified in the template element task. This might include, for example, an employee relationship between a person and a company, a family relationship between two persons, or a subsidiary relationship between two companies. Extraction of relations among entities is a central feature of almost any information extraction task, although the possibilities in real-world extraction tasks are endless." In general good TR scores reach around 75%. TR is a weakly domain dependent task.

2.3.5. Template Production

The TE task builds on NE recognition and coreference resolution, associating descriptive information with the entities. The format is an arbitrary one; it is essentially a database record, and it could just formatted for SQL store operations, or reading into a spreadsheet, or (with some extra processing) for multilingual presentation. As in NE recognition, the production of TEs is weakly domain dependent, i.e. changing the subject-matter of the texts being processed from financial news to other types of news would involve some changes to the system, and changing from news to scientific papers would involve quite large changes.

2.4. Components of information extraction

The major components of any information extraction task include analysis of words morphology, part-of-speech tagger, and syntactic analysis.

2.4.1. Morphological analysis

One of the fundamental computational tasks for a language is analysis of words morphology, where the goal is to derive the root and grammatical properties of a word based on its internal structure. Morphological analysis, especially for complex languages like Amharic, is vital for development and application of many practical natural language processing systems such as machine readable dictionaries, machine translation, information retrieval, spell-checkers, and speech recognition [38].

Many information extraction systems for languages with simple inflectional morphology do not have a morphological analysis component at all. In English, it is easy to simply list all inflectional variants of a word explicitly in the lexicon. However, for local languages in Ethiopia there are no any readymade pre NLP components that can be used for higher level NLP tasks. For languages like Amharic, with complex inflectional morphology, a morphological analysis component makes more sense. Where compound nominal are agglutinated into a single word, morphological analysis is essential. We will use morphological analyzer for our work for feature extraction purposes. The other important point should be raised here is the words morphology helps to know where the entities are derived from, for instance if we take deverbal entities most them are derived from verbs. The deverbal entities are dominantly an alternative representation of events in a text. So, that a nominal might by itself is ambiguous for the machine and rules to categorize them under events category, because of that entities are large in number and they might represent nonevent information like person name. To disambiguate them a nominal analyzing morphology is key solution.

2.4.2. Part-of-speech Tagger

POS tagging is the task of assigning lexical class marker to each word in a corpus [16]. Probabilistic part-of-speech taggers uses a lexicon as a component for automatically assigning words with appropriate part-of-speech and a central component for higher level NLP tools such as parsers, noun phrase Chunker, speech synthesis, speech recognition, information retrieval, word sense disambiguation etc. Amharic part-of-speech, although Amharic is one of the most studied languages of Ethiopia, there is no consensus as to how many POS there are for this language. The newly identified tagsets for Amharic are totally around thirty, but, the basic tags are 11 and the others are subclass of the major tagsets [14]. The POS in our work adopts a multilingual publically available tree tagger. The output of this particular component contains each word is annotated with the corresponding word class. This will be used as input for the extraction component with morphological analyzer and Gazetteers. It is one of the important features for the machine classifier component.

2.4.3. Syntactic analysis

In contrast to POS tagging, syntax analysis, also called syntax parsing, looks beyond the scope of single words. Syntax analysis identifies syntactical parts of a sentence (verb group, noun group and prepositional phrases) and their functions (subject, direct and indirect object, modifiers and determiners). Simple sentences, consisting, for instance, of a main clause only, can be parsed using a finite state grammar. Simple finite state grammars are often not sufficient to parse more complex sentences, consisting of one or more subordinate clauses in addition to the main clause, or containing syntax structures, such as prepositional phrases, adverbial phrases, conjunction, personal and relative pronouns and genitives in noun phrases. Basically syntactic analysis is used to parse a sentence when it is needed for higher level NLP applications including machine translation.

2.5. Approaches to information extraction

The predominant category of information extraction approaches includes data-driven or statistical, knowledge based and hybrid approaches. The detailed descriptions of each approach are explained below and in chapter 3 of this document.

2.5.1. Knowledge Engineering Approach

The Knowledge Engineering Approach is characterized by the development of the grammars used by a component of the IE system by a “knowledge engineer”. Rules are developed through the help of domain experts to extract sought information from unstructured texts for the corresponding task of IE. Typically the knowledge engineer will have access to a moderate-size corpus of domain-relevant texts, and his or her own intuitions [46]. It is obviously the case that the skill of the knowledge engineer plays a large factor in the level of performance that will be achieved by the overall system. In addition to requiring skill and detailed knowledge of a particular IE system, the knowledge engineering approach usually requires a lot of labor as well. Building a high performance system is usually an iterative process whereby a set of rules is written, the system is run over a training corpus of texts, and the output is examined to see where the rules under and over generate. The knowledge engineer then makes appropriate modifications to the rules, and iterates the process. Thus, the performance of the IE system depends on the skill of the knowledge engineer.

2.5.2. Machine Learning Approach

Data-driven approaches are commonly used for natural language processing applications. These approaches rely solely on quantitative methods to discover relations. Data-driven approaches require large text corpora in order to develop models that approximate linguistic phenomena. Such event extraction techniques are not restricted to basic statistical reasoning based on probability theory, but encompass all quantitative approaches to automated language processing, such as probabilistic modeling, information theory, and linear algebra [13]. These methods focus on specific features, such as words and *n*-grams, as well as their associated weights, which are mostly determined using frequency counting algorithms. These features and their associated weights represent the input of complex clustering or classification algorithms despite their differences, all focus on discovering statistical relations, i.e., facts that are supported by statistical evidence. These discovered relations, however, are not necessarily semantically valid, as semantics are not explicitly considered, but are assumed to be implicit in the data.

2.5.2.1. Supervised machine learning algorithms

Supervised learning uses training data to induce extraction rules. Thereby, almost no knowledge about the domain is necessary, but a large set of training data has to be annotated according to the underlying structure of information to be extracted. The main bottleneck of supervised IE systems is the preparation of the training data. Most systems need a large amount of annotated documents for a particular extraction task, which also leads to the lack of portability of an IE system. However supervised learning is known to be dependent on the availability of a large amount of manually prepared training data. Even though supervised learning saves human expert time, it is still a time-consuming task to prepare the training data manually [29].

2.5.2.1.1. Naïve Bayes Classifier

Bayesian classifiers are statistical classifiers. It is based on Bayes' theorem. It assumes that the effect of an attribute value on a given class is independent of the values of the other attributes [68]. This assumption is referred as class conditional impartiality. It is one of the best text classification techniques with various applications in personal email sorting, document categorization, email spam detection, sexually explicit content detection, language detection and sentiment detection etc. It is recommended employing NB classifier when we have limited resources in terms of CPU and Memory. On the other hand when the training time is crucial factor NB is preferable because it can be trained very quickly. The theoretical background of Naïve Bayes classifier assumes independent of features. For text classification scenario we will use the tokens of the text as a feature to classify it to the appropriate class. By using the maximum posteriori (MAP) decision rule, we come with the following classifier. The following formulas are according to the work done in [69]

$$Cmap = \arg\max \ \left(\mathrm{p}(c|d)\right) = \arg\max(p(c) \prod p(tk|c)\ ))$$

Where t_k is words of the text, C is the set of classes that is used in the classification, $\mathrm{p}(c|d)$ the conditional probability of class c given text d, $\mathrm{p}(c)$ the prior probability of class C and $\mathrm{p}(tk|c)$ the conditional probability of token t_k given class C. It implies in order to find the target class in which the new instance is going to be labeled first we have to estimate the product of the probability of each word among the instance given that a particular class or probably likelihood of it multiplied by the probability of the particular class. The immediate task after calculating each of those class values is selecting the class with highest probabilities. As a solution to get rid of float point underflow of specific decimal point accuracy in place of maximizing the product of the probabilities it's better to maximize the sum of their logarithms:

$$Cmap = \arg max \left\{ \log\ p(c) + \sum_{1<K<n} \log\ p\ (tk|c) \right\}$$

After all in place of taking the class with the greatest probability values take the class which is with the greatest log mark. The decision of MAP will remain the same in case of the algorithm function is monotonic. If a particular feature does not exist in a certain class, its conditional probabilities will be zero. The product becomes zero in case of using the first decision method, when we apply the second one the product log zero going to be undefined. To get rid of such headache we will use Laplace smoothing by adding one to each count: [68]

$$p(t|c) = \frac{(Tct + 1)}{\sum_{teV}(Tct + 1)} = \frac{(Tct + 1)}{\sum_{teV}(Tct) + \beta'}$$

Where β' is equal to the number of the words contained under the vocabulary V.

2.5.2.2. Unsupervised machine learning algorithms

Unsupervised Learning systems provide only an account of the relevant information. There is no any patterns are provided by the user of the system for the sack of extraction. In this learning an annotated corpus is not used to improve the system’s level of performance. It’s difficult to realize the particular user requirement in accordance with each pattern [70]. The system expands an initial small set of extraction patterns. In general, it is not necessary to create all the components of an information extraction system using only one particular approach. It is quite possible to interchange these two approaches while building different components of the system. One of the reasons of having such a possibility is that one can never say objectively which approach is better. Both of them have their advantages and disadvantages. The advantage of a machine learning based system over the other is that it can be transferred to a different domain easily as long as specific texts and a person who can annotate them are available. But sometimes those texts are problematic or expensive to obtain or there is a lack of useful documents on which an algorithm can learn, and manual (or even machine-aided) annotation on the scale needed to provide reasonable levels of performance may be expensive. So, choosing the right approach for certain extraction task will be depend on suitable conditions like knowledge engineers or enough training data.

2.5.2.3. Semi-supervised machine learning algorithms

Using semi-supervised learning, a system learns from a collection of labeled and unlabeled data. In many applications, there is a small labeled data set together with a huge unlabeled data set. It is not good to use only the small labeled data set to train the system because it is well known that when the proportion of the figure of training samples to the number of feature measurements is small, the training result is accuracy will be affected. Therefore, the system needs to combine labeled and unlabeled data during training to improve performance. The unlabeled data can be used for density estimation or preprocessing of the labeled data, such as detecting inherent structure in the domain. In other words, the system extracts patterns from the annotated data, and labels the unannotated data automatically using the patterns. As a result, all data are labeled for the training. It saves human effort while the performance can be as good as the performance of a supervised learning technique.

2.5.3. Hybrid Approach

Scholars strongly agree that it’s hard to solely relay on either pattern based or data-driven approaches because each approach has their own constraints. Machine learning based systems score relatively minimum result in comparison with the hybrid approach. One can be the subsequent of the other in which pattern based system can be learned by applying machine learning algorithms the reverse is correct for all. In hybrid information extraction systems, due to the usage of data-driven methods, the amount of required data increases with respect to knowledge-driven systems, yet typically remains less than is the case with purely data-driven methods. Compared to a knowledge-driven approach, complexity and hence required expertise is generally high, as well, due to the combination of multiple techniques. This also often leads to higher training and possibly higher execution times [13].

On the other hand, the amount of expert knowledge required for effective and efficient event discovery is less than for pattern-based methods, because lack of domain knowledge can be compensated by using statistical methods. As for the interpretability, attributing results to specific parts of the event extraction is more difficult due to the addition of data-driven methods. Yet, interpretability still benefits to some extent from the use of semantics as in knowledge-based approaches.

2.6. Evaluation metrics

The idea behind the MUC is not only the competition between different research groups in information extraction problems, but, also evaluating their systems performance. The baseline they has been using as evaluation metrics are the standard IR metrics such as precision, recall and F-measure. In any information extraction tasks the evaluation are expressed based on the notion of false positives, true positives, false negatives and true negatives. Scholars define that the values which is extracted correctly are true positives whereas false positives are wrongly extracted values. On the other hand true negatives refer values, which is relevant but not extracted and false negatives (false drop) refers the values which is not important and not extracted. In general term in IE discipline we categorize them as type one and type two errors. To describe the concept of each standard evaluation metrics we need to parameterize them as follows. All the following standard metrics formulas are described in the work of [66].

Precision: is the proportion of instances that are correctly extracted which a true positive instance.

$$precision(p) = \frac{\mathrm{tp}}{\mathrm{tp+fp}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. (2.1)$$

Where t_p refers true positives while f_p refers false positives

Recall: refers the proportion of number of correctly classified instances over the total number of the test set.

$$Recall(r) = \frac{\#\mathrm{tp}}{\#\mathrm{N}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(2.2)$$

Where t_p refers true positives while N refers total number of instances

F-Measure: is an optimization criterion which often is used for tuning the threshold in binary decision, which is defined in terms of precision, and recall, which is the harmonic mean of precision and recall.

$$F - measure = \frac{(\beta 2 + 1) * \mathrm{precision} * \mathrm{recall}}{(\beta 2 * \mathrm{precision}) + \mathrm{recall}} \ldots\ldots\ldots\ldots\ldots.(2.3)$$

Where β parameter allows differential weighting of recall and precision, if β is greater than one, then precision becomes more important than recall. On the other hand, if β is less than one, then recall becomes more important than precision. The other possibility is if β =1 then precision and recall becomes equal, and the above equation optimized to

$$F - measure = \frac{2 * \mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \ldots\ldots\ldots\ldots\ldots.(2.4)$$

CHAPTER THREE

3. RELATED WORK

Event extraction has recently gained popularity due to its wide applicability for various NLP applications. Most event extraction systems support English and European language texts from different domains using variety of techniques. For many Semitic languages a research need is felt. Event and temporal expression extraction for Amharic has not been done still, therefore our work is the first in this particular IE application. Due to the variation of the language structure the existing techniques and tools applied for other languages might not be used without relevant selective criteria and modification for our work. In this work we reviewed various data-driven, knowledge-driven, and hybrid approaches for event and temporal expression extraction. The works done in the reviewed literature evaluated on a set of qualitative dimensions, i.e., the amount of required data, knowledge, and expertise, as well as the interpretability of the results and the required development and execution times.

3.1. Data-driven Approach

Data-driven approaches develop models of text corpora that approximate linguistic phenomena. Such event extraction techniques are not restricted to basic statistical reasoning based on probability theory, but encompass all quantitative approaches to automated language processing, such as probabilistic modeling, information theory, and linear algebra [13]. These methods focus on specific features, such as words and *n*-grams, as well as their associated weights, which are mostly determined using frequency counting algorithms. These features and their associated weights represent the input of complex clustering or classification algorithms, despite their differences, all focus on discovering statistical, i.e., facts that are supported by statistical evidence. There are three popular approaches exist under data driven techniques such as supervised, semi supervised and unsupervised approaches. The supervised learning techniques typically produce new events, based on the given labeled examples. It deduces event properties and characteristics from training data to generalize unseen situations. Some of supervised techniques for learning relations includes support vector machine, decision trees and neural networks require a large amount of data to be trained.

The authors in [41] propose a state-of-the-art supervised machine learning approach for extracting events out of Arabic tweets. The proposed approach focuses on four main tasks: Event Trigger Extraction, Event Time Expression Extraction, Event Type Identification, and Temporal Resolution for ontology population. Their pipeline of processes starts with common preprocessing including feature extraction and classification tasks and finally ends with by feeding up stage to populate the ontology. Significant features used for the classifier are event trigger, event time and event type. Features extracted are morphological, syntactic, and semantic and word features. The overall annotated datasets used are 2000 sentences. Their scores shows that the proposed approach performs significant result in tasks T1: event trigger extraction F-1= 92.6, and T2: event time expression extraction F-1= 92.8 in T3: event type identification Accuracy= 80.1. The authors claim that the third task is relatively worse than the works done by similar authors using unsupervised rule-based approach.

The work in [18] detects French and English TimeML Events in using combination of different supervised machine learning algorithms such as conditional random field, decision tree and k-nearest neighbor including language models. Event extraction under the tempeval challenges framework follows the ISO-TimeML language specification; the annotation includes event expression, temporal expression, temporal relation, aspectual, alink and other language markers. The authors in this paper used it as a baseline. Afterwards TimeML annotated data are provided to train their system, which are then evaluated on separate test set. After the training phase, the inferred classifiers can be used to extract the events from unseen texts by assigning the most probable label to each word with respect to its context and features. The features used in their work categorized as internal features like: word-form, lemmas and part-of-speech in addition external features such as wordnet are included.

The results reported in this work shows a combination of CRF-KNN outperforms over CRF-DT. The metrics used are usual precision, recall and f-measure. Their work is tested on English and French corpora and scores different results due to the impact of corpus size and lexicons. Totally they have been used category of all events, nominal only and nominal without states score result of F-scores 86%, 62% and 57% respectively this result is only for the best performance approach in comparison with the other combinations. This paper has strong points like identifying the performance of combination of methods as well as the features impact in each method, but they solely relay on machine learning techniques. The result of their work shows for nominal only and nominal without state category scores minimum result. This is due to the power of the machine learning algorithm to detect nominal events without any help of rules or dictionaries to disambiguate the nominal events.

In [4] the Authors propose machine learning based techniques to extract event including places, dates from the unstructured text. In their work the first task is file classifier the user provides MS-word or PDF file format then their system communicate with external component like PDFOne (PDF SDK) and Apache POI (Poor Obfuscation Implementation). Both tools are java API in which the former enables easily manipulate PDF files like split, annotate and mark etc. The later helps to manipulate the MS-word file format to .Word document can be considered as very long single text buffer. HWPF API provides "pointers" to document parts, like Sections, paragraphs and character runs. After that tokenization is applied and pattern analysis is done by identifying the sentence boundary and applying different rules to identify the date expressions. The next step is feature extraction; in which tokens lexicon classes are identified they used Stanford POS tagger. Finally they applied NLP rules to identify places and tracks from the event document. They extend their work by doing query based dialogue management to allow making interaction with the user for the required dialogue that will answer the basic WH questions. The evaluation metrics used in this paper are the popular evaluation metrics for IE such as precision, recall and F-measure.

The Authors in this paper has done important things like query extracting event related arguments like places beyond events and they extend their work to query based dialogue management apart from their objective for the user instructiveness. The author didn't address in their work is event and temporal information relation extraction, because it will be very helpful in their extended work. The other thing author uses static rules to extract event arguments.

The works in [3] propose state-of-art for event and temporal information extraction in English documents under the SemEval-2016 Tasks in which they identify events, time expressions and temporal relations among them. The approaches in this paper are relatively better and score good results as well as they cover relevant tasks to be done in this particular IE application. The methods used in this paper is using supervised machine learning algorithms including, Conditional Random Field, Support Vector Machine and Deep Neural Networks. The result of each algorithm is relatively the same performance. Among the participants in this challenge the top performing systems used hybrid approaches with machine learning techniques and rule based tools. In general term the tasks done in this paper are sequence labeling, classification and relation extraction. They used CRF ++ suite tool by applying features like term feature, POS tags(using Open NLP Tagger), Orthographic features, Stop words and Train Events Dictionary, to identify event and time spans. To identify Event and time attributes they train support vector machine for each of the attributes to classify in to their respective classes. They used word representations or word embeddings as the features for training the SVM classifier. The word representations are generated based on the co-occurrence count modeling using Stanford Glove tool. Finally they try to identify the Document time relation and narrative container relation using temporal link to order events these links are only provided for the events that happen within a temporal bucket. The next separate approach for all the above subtasks of this challenge used in this paper is using deep neural network to take an advantages in the advancements. This particular approach of the authors work in a manner given an input sentence and output tags the neural network learns the weights of nodes in each word among the sentence. The input words are represented as word embeddings which are same as that of word representations used for SVM classifier in their first approach. They trained separate neural networks for each of the tasks that have been done by their primary approach.

Another work in [36] one of tempeval challenge of 2016 to extract temporal information from unstructured clinical narratives using machine learning approaches such as support vector machine and conditional random field. The tasks mainly focus on six subtasks such as span detection, attribute classification, EVENT span detection and attribute classification, relation between each event and document creation time classification and narrative container. They build UIMA module to segment the clinical notes and cTAKES clinical pipeline was used to extract lexical and syntactic features. Their work was based on two standalone tests using CRF and SVM. The standard SVM library, LIBLINEAR package within ClearTK was used to train all linear classifiers with default settings in their work, whereas , CRFsuite used for the sequential classification, TIMEX3, EVENT and DocTimeRel subtasks in phase 1, and CRF++ for the DocTimeRel subtask in phase 2 of their work. They achieved competitive results in each subtask with an F1 75.4% for TIMEX3, F1 89.2% for EVENT, F1 84.4% for event relations with document time relation, and F1 51.1% for narrative container (CONTAINS) relations. Even though the result reported for some particular task is significant, but for TIMEX3 and narrative container relations their result is incompetent with others work result that has been done at the same time with different approaches by applying the machine learning techniques along with some heuristics.

3.2. Knowledge-driven approach

Knowledge-driven event extraction methods often use predefined (learned) patterns expressing expert knowledge rules. Their TM procedures are hence inherently based on linguistic and lexicographic knowledge, as well as on existing human knowledge regarding the content of the texts to be processed.

[2] The author in this paper addresses the overview of event extraction from texts including its wide application, existing techniques and to aid how to choose a particular event extraction technique depending on the user, available content, and the scenario which is going to be applied. In this paper events are described well as knowledge's extracted by a means of text mining, in which it's a complex combination of relations linked to a set of empirical observations from the set. There are different domains described in their work in which event extraction will be applied like biomedical domain, risk analysis, algorithm trading, financial market, dividend announcement and economic events. Due to its potential of wide application there are methods studied and compared by their work. Techniques discussed here are data driven, knowledge base and hybrid approaches. We will not discuss their detail here because we will talk on Section 5 of this docuemnt. After they define the constraint in each technique, they select criteria like the amount of required data, as required domain knowledge and expertise (Know. and Exp., respectively) and the interpretability of the results to summarize them and suggest the best techniques. Finally as general guideline they conclude that techniques for casual users that prefer an interactive, query driven approach to event extraction, assuming domain knowledge and expertise to be readily available. Users can easily specify patterns in a language that is close to their own natural language, without being bothered with statistical details and model fine tuning. On the other hand, users like (academic) researchers would benefit from both hybrid and data-driven approaches, as these are less restricted by, for example, grammars.

We have learnt many things from this paper particularly the existing techniques for Event extraction, so that their work is appreciable. The other thing that might seem subjective in this paper is the techniques selection criteria because the choice depends on the scenario at users hand and his capability as well as his interest apart from achieving good performance of the desired system.

Another work in [35] designed a system to extract temporal information using regular expressions and apply normalization functions to them. The objective of this paper is extracting the extent of the time expressions in a text as defined by the TimeML timex3 tag. In addition, determine value of the features type and Val. The possible values of type are time, date, duration, and set; the value of Val is a normalized value as defined by the timex2 and timex3 standards. Example of the temporal expression are **on March 28/በመጋቢት 28, last week/ባለፈው ሳምንት, yesterday/ትላንት, tomorrow/ነገ, last month/ባለፈው ሳምንት,** etc. In order to recognize temporal expressions, a hierarchy of regular expressions is used. To normalize the hierarchy of regular expressions was extended to be a hierarchy of three-tuples consisting of a regular expression, the category and a function which transformed the matched expression to the normalized form using TIMEX3 tags. The authors claimed that their work was good in comparison with other regular expression based systems, but further analysis of sentences and maximizing window size yields significant improvement. The problem here is as other works suggests we cannot totally relay on regular expression based systems because, since rules are handcrafted and language dependent it will not be applied for other languages, needs linguistic knowledge, and the result is also not satisfactory.

The system in [9] describes usage of semantic parsing to collect automatically events from text and convert them into the LODE model. The system also automatically links extracted event properties to the external resources like DBpedia. Some of the event models are listed in this paper such as LODE, SEM they share common features for representing event participants, place and time and represent them in RDF vocabularies. The initial problem addressed by this paper is that event mentions of the above event arguments from their source are rarely comply with their representation. So that events are detected and extracted, to do that this paper has proposed an approach semantic role labeling, which is a generic technique to parse predicate argument structures. The pipeline they designed follow four stages such as Semantic parsing of Wikipedia (SRL), which parse texts using framework. Event selection: argument identification and property extraction, which associates the predicate–argument structures extracted by the first module and relates them to a restricted set of Verb Net roles. Disambiguation and linking of the time and location phrases to external resources, that associates agent, time, and location phrases to GeoNames and DBpedia entries and finally Mapping of the predicate–argument structures onto an event model, which maps the structure to the LONDE event model. The semantic representation of sentences includes semantic roles and Event models, popular ones explained Framenet and Popbank applied here. The s, is core parser their system uses a high-performance multilingual semantic role labeler. To map predicates to event they simply link Popbank and VerbNet by doing some mapping staff other than rules. Finally to select event proposition they built event set from the complete proposition output produced from Wikipedia and conversions are applied from propositions to event models and the extracted events are saved to file in Notation three format.

In this paper the works has been done is appreciable because semantic analysis helps to solve the headaches faced by syntactic analyzers as well as the output is interpretable, so that it can be extended for further application easily. The problem raises slowness of the techniques when it will be applied in large corpus. The semantic role labeler is also criticized by other researchers because of its prone to errors especially with phrases involving proper nouns and adjuncts.

Another work in [37] designed a rule based extraction and normalization of temporal expressions. They used HeidelTime, which is totally a rule based system using regular expression to extract temporal expressions. The overall system architecture includes a pipeline of at the beginning the UIMA framework for document processing by accessing the documents using Collection Reader from a source and initializing a CAS object for each document. Next to that extraction and normalization tasks are performed on the middle, where the normalized value represents the temporal semantics of an expression as it is specified by TimeML.

Example: extraction and normalization resources for months.

Expression   *reMonth* = “(. . . |June |July|. . .)

Normalization norm*Month* (“June”) = “06”

At the bottom there is functionality of HeidelTime and postprocessing, the last post-processing step is to remove all extracted timex annotations that are invalid. In the result Section they achieved a competitive result of highest FScore (86%) for the extraction and the best results in assigning the correct value attribute, i.e., in understanding the semantics of the temporal expressions.

The authors in [39] presented a three-step framework providing techniques and solutions for extracting and aggregating temporal events from texts. In their approach, first they present a configurable framework for extracting event-related facts from the web. Second, they suggest suitable methods for each step: including document retrieval, information extraction, and time-aware aggregation. Third, they apply our framework to a real-life case study extracting disaster events of earthquake from web sources.

The overall flow of their framework perform each subtask as, the retrieval module takes an event as input and returns a set of event-relevant HTML documents. Then extraction module takes a set of documents as input and initially extracts for each document its structure, i.e. the title, the description and the content. Finally the fusion module takes as input a set of temporal aligned facts and returns condensed review overtime. For the retrieval case to find relevant document the query builder and document filter options used. To extract facts pattern matching techniques has been used. In order to extract temporal expressions regular expressions along with normalization has been used. Merging extracted facts with respect to the timeline, this part of the work achieved by applying outlier detection with a set of time-aware strategies to deal with these inconsistencies adequately, providing a consistent view on events. The other task accomplished in this paper is document fusion classified as intra and inters document fusion of documents, which pertains valuable facts. Finally framework configuration, performed to minimize the difference between the framework's output and the Wikipedia reference. The authors' claims in their work this framework performs well to extract facts and merging them with minimal error rates by considering the cost manual data gathering.

The Authors in [40] propose Knowledge-based Approach for Event Extraction from Arabic Tweets. There are three subtasks under their work, event trigger extraction, event time extraction and event type identification. The event expression includes important event arguments, which are event agent, event location, event trigger, event target, and event product and event time. Their dataset were collected by the help of twitter streaming API and preprocessed through AraNLP java based package. Moreover after events extracted visualization services like calendar, timeline supplied through the help of ontological knowledge bases. The knowledge base adopts a linked data approach where entities are linked through a set of knowledge bases like Wikipedia or DBpedia. Finally event extraction and disambiguation holds three sub-processes such as Extracting Event mentions, Named entity recognition and disambiguation, and temporal resolution. Rule-based approach used to extract those event mentions and temporal expressions. Their experiments results show that the approach has an accuracy of, 75.9% for T1: event trigger extraction, 87.5% for T2: Event time extraction and 97.7% for T3: event type identification. The Authors in this work claims that applying this kind of domain dependent approach to extract events from tweets scores significant results.

3.3. Hybrid Approach

The third approach is a combination of data driven and knowledge driven approaches. In hybrid event extraction systems, due to the usage of data driven methods, the amount of required data increases with respect to knowledge-driven systems, yet typically remains less than purely data-driven methods [13]. Compared to a knowledge-driven approach, complexity and hence required expertise is generally high, due to the combination of multiple techniques. This also often leads to higher training and possibly higher execution time.

In [25] the author implemented event extraction system for biological domains using full parser and preprocessor with a large scale general purpose grammar approach. The parser converts different sentences that describe the same event into argument structure for the verb representing the event and its argument such as subject and object. Afterwards event information is then extracted by domain-specific mapping rules from argument structures to frame representations. The general purpose grammar simplifies the syntactic analysis tasks. They also add two preprocessors that resolve the local ambiguities in sentences to improve the efficiency, the term recognizer and shallow parser. The first one enables to identify domain specific technical terms and classify them into semantic classes, whereas the second one reduces the lexical ambiguity by using local constraints to avoid more unlikely part of speech and try to make dependency structure that can be constructed locally. Finally they used the postprocessor that receives and handles the result of the parser.

The overall system design contains a flows at the beginning the argument extractor which accept the input text and next to it the frame extractor is applied to construct user defined frame representation from the argument structure. They used a named entity annotated corpus, for argument structure extraction total of 179 sentences parsed from those 31 extracted uniquely, 32 extracted with ambiguity and 70 not extracted total 133, from this with the disambiguation and the postprocessor that enables the system to use the partial results, 99 (74%) argument structures in total are expected to be extracted. Finally they conclude that the approach they applied is convincible for the system in this particular domain. This paper pertain strong points like applying the disambiguation method and the preprocessor components to achieve optimal results, but as other scholars argue that and from the results of this paper we understand applying full parser will not yield good result over shallow parser for IE tasks like event extraction and others.

In [45] the authors propose a full temporal information extraction approach, which recognizes a wide range of temporal expressions and runs probabilistic inference to extract point-wise constraints on the endpoints of event-intervals by making full use of transitivity. The overall overview of their work follows a flow of preprocessing including, parsing, semantic role labeling, and GUTime, moreover classification based on given features have been done through pretrained probabilistic models. For the case temporal relation they used markov logic network (MLN), which is a set of weighted first order formula. In their result and discussion Section the author compares their result with other three different temporal extractor systems, so that their system slightly increased precision and substantially improved recall.

Another work presented in [33] under clinical tempeval challenge of 2015, the challenge includes TIMEX3 and event span detection, TIMEX3 and event attribute classification, document relation time and narrative container relation classification. The authors in this work participated in all tasks by applying different approaches for each subtasks. The methods used for each tasks consequently ClearTK support vector machine pipeline, conditional random field and rule based approaches separately. Some important attributes with lexical features were identified and used in their work like: type, degree, polarity, and modality. The possible values of event attribute and timex3 attribute are specified. Finally in their result and discussion Section the authors claim that their benchmarking rule based tools are proved insufficient, rather applying machine learning approaches along with it yields optimal result.

In [15] the authors used state-of-the-art tools for temporal expression and event recognition and bring them together to form an openly-available resource within the GATE infrastructure. GATE-Time provides annotation in the form of TimeML events and temporal expressions complying with this mature ISO standard for temporal semantic annotation of documents. Major advantages of GATE-Time is that it relies on HeidelTime for temporal tagging, so that temporal expressions can be extracted and normalized in multiple languages and across different domains. They try to compare the popular temporal tagger based on the domain sensitiveness like SUTime and HeidelTime, in which the authors work shows due to its normalization quality and multilingual behaviors HeidelTime is chosen as better than the former.

The HeidelTime Gate wrapper performs some linguistic preprocessing. When we incorporate it with the UIMA framework which resembles the Gate wrapper the feature will not do anything, it uses user provided annotations and just adds TIMEX3 annotations. In addition to tokens, sentences, part-of-speech HeidelTime requires extra information to extract and normalize the temporal information, those requirements include language and text type like news-style or narrative and the document creation time are needed. The authors also describe how they extract events also suggests the popular tools (EVITA, ATTI, TIPSem) for event extraction. The authors' idea is identifying entities and relation between them, so that the event will be the relation. Gate event recognizer has been used in their work which includes event gazetteer, verb phrase Chunker and event recognition grammar. Since events in a sentence include verbal predicate, noun phrase, adjective noun combinations and event referring nouns, the major ones are explained well verbal predicate and nominal nouns. They applied PAUM algorithm for verbal event extraction and classification. Finally they evaluated their approaches on standard corpus they conclude that gate time achieves state-of-the-art results.

The authors of this paper did appreciated work they used state-of-art approaches and includes important tasks like relationship and attribute extraction in comparison with other literatures approach and results. The authors in this paper solely concerns statistical approaches to tackle all the subtasks in this challenge, but as they describe in their introduction Section applying some heuristic and rules together with the machine learning technique yields better result. Based on this assumption gained applying rules for some of the components that needs some analysis will be better. The second approach in this paper scores less than their first approaches for event and temporal relation extraction, it seems has nothing add value for the system performance other than adding complexity for users and makes their work unrealistic.

SemEval-2016 Task 12: Clinical TempEval challenge [28] their work has two phases in the first phase where they identified text spans of time and event expressions (sequence labeling) in clinical notes and in the second they predicted a relation between an event and document creation time(classification tasks). The methods applied here for the first phase they used vanilla recurrent neural network that incorporates word embeddings and for the second stage a combination of date canonicalization and distant supervision rules. The RNN trained in three separated tasks in the first phase character level, word level and POS tagger, whereas the DeepDive build domain knowledge into applications using a combination of distant supervision rules, which use inference rules that use factors to define relationships between random variables. The results reported in this paper are for three different experiments timex3 span, event span, and EVENT Document Creation Time Relation extraction, they score consequently values of f-measure 79%, 90% and 74%. The author in this work achieves good result in comparison with the previous works aiming at reducing over-engineered pipeline extraction techniques rather relay on deep learning approaches, but it seems to lose performance relative to the first component when they solely applied the second component.

3.4. Information extraction and related works on Amharic text

There are different works on general information extraction model and related works for Amharic text [29, 46, 47, 48] by Addis Ababa University Graduate school students.

The works by Getasew in [47] propose a model for Amharic Text information Extraction. Their model includes components such as document preprocessing, text categorization, learning and extraction and post processing as its main components. For learning and categorization module they used popular machine learning algorithms, Naïve Bayes and support vector machine classifier algorithms. In the result and discussion Section of their work a total of 1200 news texts used for training and testing purpose. Moreover different scenarios considered during the evaluation by adding features and minimizing features for the classifier and finally they showed that SMO algorithm correctly classified 94.58% of instances, which outperforms over the Naïve Bayes classifier that yields result of 92.8%. Even though the authors work used state-of-art machine learning approaches, their work solely focus on numeric and named data extraction the other thing it seems insignificant is their dataset is domain specific the news text talks about single issue.

The other work presented by Bekele in [29] is general information extraction system from Amharic text using Knowledge-poor approach. The system developed using GATE text processing environment using knowledge-poor approach on existing infrastructure domain. In their approach they used simple rules and gazetteers list for entity identification. The flow of their approach passes preprocessing, extraction and post processing. Other than the preprocessing the second stage of their work is the main task of this paper which performs named entity recognition, coreference resolution and relation extraction and extract relevant text. In the result and discussion part of this paper they showed that a total number of 24760 instances used for training and testing purpose and they tested their work separately as named entity recognition and overall extraction system and finally they score a result of 81.1% for the former and 81.7% for the later. Even though the authors claim that their work performs well, the extracted information doesn't seem to have that much usage as the facts which describe about entities are not parts of the research work.

Another work by Sinatyehu in [46] proposes for local language Amharic, which is rule based approach to extract information from Amharic vacancy announcement texts. The author tries to extract candidate texts from the news document. The overall flow of the system possess simple architecture at the top as usual there is preprocessing stage and at the middle their major task align, which part-of-speech tagging and candidate txt selection from small gazetteers and candidate text extraction finally post preprocessing is there at the bottom. Sintayehu's work extract candidate named entities which match along with the gazetteers using rule based algorithms. In the result and analysis Section of the authors work they used 116 Amharic vacancy texts with total of 10,766 words and they score overall F-measure of 71.7%. Even though this result is not good in comparison with other scholars result of foreign languages like English and French it's appreciable due to consideration of linguistic characteristics, resources and state-of-art approaches done for Amharic before to be used as a benchmark.

Another work presented by Atelach et al. in [49] proposes machine learning approach for Amharic text classification. The authors in their work clearly analyze the impact of stemming in text categorization for highly inflectional morphologically rich languages like Amharic is high. Components used in their work include morphological analyzer, stemmer and part-of-speech tagger. The main task of the author is classification three representations schemes used for the classifier; they used bag-of word approach represented as a vector space model. That is, each article in the corpus was represented as a (sparse) binary vector and where each position in the vector corresponds to a specific unique word that occurs in at least one of the news articles in the corpus. If an article contains a word, then the vector position corresponding to that word would have the value 1, else it would be 0. The other representation is stemmed version of the text to represent each document. The third representation, they used only the nouns from the full text of each article and represent it in the form of a binary vector. In their result and discussion Section the authors setup the experiment on RDS (Amharic rule discovery system) on 10 fold cross validation with stemmed version of 69.39%. The authors claim that giving emphasis on stemming and morphological analyzer yields a competitive result of text classification for morphologically rich languages like Amharic. See the summary in **(Appendix A)** of this document.

CHAPTER FOUR

4. AMHARIC LANGUAGE STRUCTURE

This chapter discusses the language structure and common characteristics which needed for the development of ETIEA system.

4.1. Introduction

Amharic is a Semitic language, related to Hebrew, Arabic, and Syriac. Next to Arabic, it is the second most spoken Semitic language with around 27 million speakers [38]. Also for a long period, it has been the principal literal language and medium of instruction and school subject in primary and secondary schools of the country. Moreover, it is the working language of the Ethiopian Federal Government and some regional governments in Ethiopia, most documents in the country are produced in Amharic. There is an enormous production of electronic and online accessible Amharic documents.

Amharic language is moderately expanded both when spoken and written. Compounds and loan-words haven't standard spelling in Amharic. It has a complex morphology, where nouns and adjectives are inflected for gender, number, definiteness, and case. In spite of the relatively large number of speakers, Amharic is still a language for which very few computational linguistic resources have been developed and very little has been done in terms of making useful higher level Internet or computer based applications. It is generally believed that applications such as information retrieval, question answering and machine translation could benefit from the existence and availability of basic tools such as, morphological analyzers or part-of-speech taggers and syntactic parsers. However, since so few language processing resources for Amharic are available, very little is known about their effect on retrieval or classification performance for this language. It has been argued that event extraction can improve temporal reasoning performance of question answering system. As a result event and temporal information extraction for Amharic needs linguistic analysis and preliminary NLP components that has been done for Amharic as such for other information extraction tasks.

4.2. The Amharic writing system

Amharic uses a unique script (shared with Tigrinya), which in contrast to Arabic and Hebrew is written from left to right. The script is commonly known as “Ethiopic script”, “Ge’ez” or fidel [44]. Ge’ez, which belongs to the class of Semitic languages, was the language of literature in Ethiopia in earlier times [52]. The ancient Sabaean script is in turn attributed to the source of the Ge’ez script. However, as explains, the numbers of symbols in the original Sabaean script and their shapes have changed into Ge’ez later on into Amharic. Moreover, some new symbols have been added to Amharic. The current Amharic writing are combination of all Ge`ez fidel and some added characters. In modern written Amharic, each syllable pattern comes in seven different forms (called orders), reflecting the seven vowel sounds. The first order is the basic form; the other orders are derived from it by more or less regular modifications indicating the different vowels. The alphabet is written from left to right, in contrast to some other Semitic languages. It consists of 33 consonants, giving 7*33=231 syllable patterns, or fidels. In addition to the 231 characters, there are other non-standard alphabets which contain special features usually representing labialization. Each alphabet represents a consonant together with its vowel. The vowels are used to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change their order.

4.3. Amharic punctuation marks

According to the works in [71] punctuation referred as is the usage of conservative signs and space as an indicator for the thoughtful and the correct reading of handwritten or printed texts. Other linguistic definition of punctuation refers it as the practice introducing points or small marks into manuscripts; in order to support clarification, dissection of text into sentences using such marks. The punctuation rules vary according to the language, time; register and location are constantly evolving.

In Amharic, there are different punctuation marks used for various purposes. In the old scripture, a colon (two dots ፡) has been used to separate two words. These days the two dots are replaced with whitespace. An end of a statement is marked with four dots (አራት ነጥብ ።) while ነጠላ ሰረዝ (፣ or ፥) is used to separate lists or ideas just like the comma in English and (ድርብ ሰረዝ ፤) is used as a semicolon in English. The question and exclamation marks have recently been included in Amharic writing system [47]. In Amharic, numbers can be represented using either the symbols of Arabic number system or the symbols of the Ethiopic number system or using words and symbols of the Arabic number system.

Table 4.1 Amharic number representation

Arabic	Ethiopic	Alphanumeric	Arabic	Ethiopic	Alphanumeric
1	፩	አንድ	20	፳	ሃ ያ
2	፪	ሁለት	30	፴	ሰላሳ
3	፫	ሶስት	40	፵	አርባ
4	፬	አራት	50	፶	አመሳ/ሃመሳ
5	፭	አምስት	60	፷	ስለሳ
6	፮	ስድስት	70	፸	ሰባ
7	፯	ሰባት	80	፹	ስማኒያ
8	፰	ስምንት	90	፺	ዘጠና
9	፱	ዘጠኝ	100	፻	መቶ
10	፲	አስር	1000	፼	ሺህ

4.4. Amharic Word Classes

According to Baye in [20] Amharic words are categorized under five basic classes using the morphology and position of the word in Amharic sentence as criteria. These five categories are ስም (noun), ግስ (verb), ቅፅል (adjective), ተውሳከ ግስ (Adverb) and መስተዋድድ (preposition).

- **Noun**: are words that are given to name or identify any of categories of things, people, animal, places or ideas or a particular of one of these entities. A word will be categorized as a noun, if it can be pluralized by adding the suffix ኦች/ ዎች ("owch") and used as nominating something like person and animal. It is used as a subject in a sentence. Pronouns, which were considered as independent category in the previous works by the linguistics professionals recently its categorized under nouns by considering the unique feature of the language in relative to others other than adopting the other language structures. The following are some of the pronouns in Amharic ይህ , ያ , እሱ, እስዋ, እኔ , አንተ, አንቺ…; quantitative specifiers, which includes አንድ, አንዳንድ, ብዙ, ጥቂት, በጣም…; and possession specifiers such as የእኔ , የአንተ, የእሱ.
- **Verb**: any word which can be placed at the end of a sentence and which can accept suffixes as /ህ/, /ሁ/, /ሽ/, etc to indicate subject markers. Which is used to indicate masculine, feminine, and plurality is classified as a verb. Verb expresses accomplishment of an action and used to close the sentence. For example in a sentence "የአቤል እናት ድጋሚ ወለደች" the word "ወለደች" is verb since it appears at end of the sentence and closes the meaning of sentence.

- Among the seven Amharic writing symbols order, majority of the verb words use the first order Amharic writing system
- Verbs can use 'እየ-' prefix morpheme
- Verbs can change their last symbol to the Amharic seven order writing system. Finally this changed verbs may take '-አል' suffix morpheme

Similar to nouns and adjectives verbs also derived from Verbal Roots by affixing the vowel ኧ, Verbal Stems by affixing morphemes and compound Words of stems with verbs.

- **Adjective**: those are words which describe nouns or pronouns to denote quality of a thing; that is, it specifies to what extent a thing is as distinct from something else. It will come before a noun to qualify a noun with some form of size, kind and behavior. Example ቀላል፤ ትንሽ፤ ለብቻ፤ ተመሳሳይ ፤መጨረሻ
- **Adverb**: refers a word or phrase that modifies the meaning of an adjective, verb, or other adverb, expressing manner, place, time, or degree. Some of the adverbs in Amharic include ሁልጊዜ፤ ለምን፤ ማን፤ ምናልባት፤ምክኒያቱም፤ ስለዚህ ፤ ተቀድሞ ፤ በመካከል፤ ይልቅ ፤ ደህና ፤ደግሞ ፤ የተገለበጠ ፤ እንድሁም etc.
- **Preposition**: are classes of words that express spatial or temporal relations (ውስጥ፤ ከታች፤ በፊት) or marks various semantic roles like (ለ እና ከ)

4.5. Normalization

Normalization is the task of altering text into a single recognized form that it might not had before [29]. Normalizing text before processing enables for separation of concerns, since input is guaranteed to be reliable before operations are performed on it. It requires being aware of what type of text is to be normalized and how it is to be treated afterwards; there is no all-purpose regularization procedure.

Unlike other Semitic languages in Amharic writing system there are characters with the same pronunciation but different symbols which are called homophones. The letters such as አ, ኣ, ዐ and ዓ; ሠ and ሰ; ሀ, ኀ, ሃ, ኻ, ሓ, ኃ and ሐ, ጸ and ፀ are examples of characters with the same meaning and pronunciation but different symbol. Example: the word "Hailemariam" ሀይለማሪያም it could have different forms like ሐይለማሪያም ፤ ሃይለማሪያም ፤ ሓይለማሪያም ፤ ኀይለማሪያም and ኃይለማሪያም. Therefore, to handle the variety without affecting the sense of the word the above different forms of the word ሀይለማሪያም must be normalized into ሀይለማሪያም by changing the first character of a word which makes easy to handle. Therefore, these characters should be normalized.

4.6. Amharic Verb Morphology

Amharic has a complex inflectional morphology, particularly for verbs, employing not only prefixes and suffixes but also modifications of the typical Semitic consonantal root-and-pattern type. The verb is inflected for voice, tense-aspect-mod (TAM) and person. The verbal complex may also contain object markers. There are affirmative and negative conjugations, and different conjugations for verbs in main clauses and subordinate clauses. The verb root is in most cases triconsonantal, but it can be bi-consonantal or tri-consonantal [59]. The citation form is the third person masculine, singular, of the perfect tense. In addition to the affixation, reduplication, and compounding common to other languages, in Amharic, as in other Semitic languages, verb stems consist of a root + vowels + template merger (e.g., *sbr* + ee + CVCVC, which leads to the stem *seber,* sebabere, 'broke'). The template represents tense, aspect, mood, and one of a small set of derivational categories: passive-reflexive, transitive, causative, iterative, reciprocal, and causative reciprocal [54]. Depending on the information the verbs describes in a sentence each lexeme categorized under tense mod of perfective, imperfective, imperative and gerundive. The usual word order of Amharic is subject-object-verb (SOV). However, if the object is tropicalized it may precede the subject (OSV). Noun phrases are head final with adjectives and other modifiers preceding their nouns. One of the interesting characteristics of Amharic verb may also have a suffix representing the person, number, and gender of a direct object or an indirect object that is definite, which can be considered as clitics in other Semitic languages.

4.7. Event and temporal information analysis in Amharic documents

Text documents contain different type of event and temporal information. We try to define the meaning of events as it covers a term for situations that happen last for a certain time. Events in a text most of the time refers the verb, in case of Amharic the verb comes always on the termination of the sentence [19]. E.g. አበበ በ 1990 ተወለደ፡፡ In this sentence "ተወለደ/tewelede" refers the verb which is an event [20], while 1990 refers the temporal information which answers the question when an event happens? .The main challenge here is since Amharic is Ge'ez script we have to convert and normalize the date expression to utilize the Heidel Time Tagger. The other temporal information will be Document Creation Time (DCT) or the last modified date of document is a simple example of temporal information in text documents. The document metadata usually have a DCT [21].

4.8. Events in Amharic text

Events can be expressed in a document with verbal predicates and arguments (e.g. the committee dismissed the proposal, “ኮሚቴው ሀሳብ ተስናብትዋለል), noun phrases headed by nominalization (e.g. economic growth, የኢኮኖሚ እድገት) adjective noun combinations and event referring nouns. Verb triggered events in a text are not ambigous unlike that of nominal e.g “አበበ በ1990 **ዓ.ም ተወለደ**” in this sentence the word “ተወለደ” is a a verb ,which is state event that tells the answer of what happend on abebe. Deverbal entities are mainly ambiguous to identify which on is an event. Basically, by definition an event is a situation which last for a moment. Having this definition, nominal can be an event e.g “ሰርግ/wedding” is a non derbial nominal event. Nominal events sometimes appear as deverbal and non-devrbal , in which devrbal entities are dervide from verbs incontrary the non-deverbial entities are not derived from verbs e.g “ ፈጽሞ” is a derbal entities which is an event deribed from verb “ፍጽም”.

4.9. Temporal Expression in Amharic text

Based on what temporal information expression refers to, there are different temporal expressions. *e.g.* a point in time or a duration. In addition, there are different realizations of temporal expressions in natural language. Depending on the realization, different types of information are required to determine the normalized meaning of an expression [50]. Unlike other entities name, place, location and event temporal information in amahric text are not as huge as other langauge features .

4.9.1. Types of temporal information Extraction

- **Date Expressions**: A date expression refers to a point in time of the granularity “day/ቀን” (e.g. “June /ሰኔ 18, 2015” or any other coarser granularity, like “month/ወርሀ” (e.g., “June/ ሰኔ 2015”) or “year/ዓመት” (e.g., “2015”). In other words these can be calendar dates (e.g. “January/ጥር 4”) and other verbal expressions which can be mapped to calendar date (e.g. “Last week/ ያለፈው ሳምንት”, “This month/ይሄ ወርሀ”, “next Friday/የሚቀጥለው ዓርብ”, or “this time/በዚህ ጊዜ”).
- **Time Expression:** A time expression refers a point in time of any granularity smaller than “day” such as a part of a day (e.g., “Friday morning/ዓርብ ጠዋት”) or time of a day (e.g., “3:30 pm/ከሰዓት”). In another words TIME is used for specific time points within a day, for instance, “4.05 AM/ጠዋት”, or can be relative “20 minutes ago/ከ 20 ደቂቃ በፊት”.
- **Duration Expression:** A duration expression provides information about the length of an interval i.e. the amount of intervening time between the two end-points of a time interval. Example of Duration expressions is “ten hours/ አስር ሰዓት”, “last 5 minutes / የመጨረሻወቹ አምስት ደቂቃወች”.

- **Set/Frequency Expression:** A set expression refers to the periodical aspect of an event, for e.g. “Every Friday/ሁልጊዜ ዓርብ”, “thrice a day/በቀን ሶስት ጊዜ”). Medical Documents like discharge summaries have various frequency terms denoted by Latin abbreviations such as, “tid (thrice a day)”, “q4h (every four hours/ በየ 4 ሰዓቱ)”.

Table 4-1 Lexical Triggers for Temporal Expression

Part-of –speech	Lexical Trigger	Non-Triggers
Noun	Day/ቀን, minute, weekend, midnight, millennium, era/ዘመን, semester/አጋማሽ, summer/ክ ረምት, the future/ የወደፊቱ, the past/ያለፈው,	Instant/በዚያን ቅጽበት, jiffy/በቶሎ
Proper name	Monday, January, New Year’s Eve,	
Time Pattern	His Birthday	
Adjective	8:00, 12/2/00, 1994, 1960s	earlier, ahead/ወደፊት, subsequent/ተከታይ,
Adverb	current, future, former, past, next,	frequent, later
Time noun /	medieval, monthly	
Adverb	currently, lately, then, next, hourly,	

CHAPTER FIVE

5. SYSTEM DESIGN AND IMPLEMENTATION

In this chapter we talk about the proposed models of the system along with the design constraints and implementation issues.

5.1. Introduction

In this Section we discuss the overall design of our system, Event and Temporal information extraction for Amharic (ETIEA). At the start of this Section we explain a bit more about our data set which is used for training and testing phases. The next Section discusses the general overview of the proposed system architecture from the perspective of the system's flow of operations. Finally, detailed explanations of each model along with subcomponents in each phase are discussed.

5.2. Datasets

Unlike other languages for Amharic there are no any standardized annotated publically available corpora like Treebank and propbank for English. Actually, the news domain is preferable data source because of its publically availability and rich source of information for any NlP applications including entity extraction, event and temporal information extraction and coreference resolution etc. In this work we used Amharic corpus which is prepared by the Ethiopian Languages Research Center of Addis Ababa University in a project called "The Annotation of Amharic News Documents". The project was meant to tag manually each Amharic word in its context with the most appropriate parts-of speech. This corpus is prepared in two forms i.e. in Amharic version (using Ge'ez fidel) and in transliterated format using Latin characters in SERA version. But we use the Amharic version one; even though it is not as easy as the Latin script, because of its spelling variations. In the preprocessing module we tried to incorporate simple normalization schemes to avoid this syntactic issue like having multiple terms, which has semantically similar meanings. The corpus has 210,000 words collected from 1065 Amharic news documents of Walta Information Center, a private news and information service located in Addis Ababa, Ethiopia. It doesn't mean that we have used all this dataset for each different model in the same manner. Basically, we have different models with their own components applied in this work; as a result we used the dataset in accordance with the models requirement.

5.3. Approach used

Based on related works we reviewed different approaches have been used throughout the state-of-art approaches of event and temporal information extraction. Researchers have their own point of view with evidence to apply a particular approach in such domain. To make it precise we don't need to go detail explanation of each class of approaches such as data driven, knowledge based and hybrid approaches, because we already state them with their pros and cons in Section 2.5 and 3. In this Section before we are going to the system architecture we need to highlight about the approaches we used in this work and the reasons behind it.

In this work we have two independent tasks event extraction and temporal information extraction. For the event extraction task machine learning classifiers, rule based and hybrid approaches are employed. In general term the class of methods covered for the event extraction task includes statistical, knowledge and hybrid approaches. Basically the methods and particular techniques for this task are supervised machine learning classifier, some preprocessing components, and some useful extraction rules, and list of gazetteers. To extract temporal information extraction we used regular expression and rule based approach. In fact rules and regular expression can be used interchangeably; to avoid confusion rules cover broader sense and it would apply a series of rules written down and coded by hand (or translated automatically from some sort of markup, meta-programming language, regular expressions, FST, etc.), whereas regular expression is a specific pattern that provides concise and flexible means to "match" (specify and recognize) strings of text, such as particular characters, words, or patterns of characters. In our case the rules used are using syntactic features and context of words in a text. The regular expression uses matching by developing complex patterns of temporal expressions. The reason behind our choice of an approach for the former task is due to linguistic characteristics, proper corpora availability, and performance issue and literature background of state-of-art approaches. Whereas, for the later task we used rule based methods because of temporal expression in Amharic text are not large in number as the other word classes. To do so, our assumption is sufficient for such kind of expressions in unstructured text. It's easy and powerful applying techniques such as complex regular expression as well as rule based approach to extract temporal information in unstructured amahric text.

5.4. System Architecture

As we described in Section two of this document, although information extraction systems that are emerged for different tasks often differ from each other in many ways. There are core elements that are shared by nearly every extraction system, regardless of whether it is designed according to the Knowledge Engineering or Automatic Training paradigm. The general system architectures of an IE system was defined in MUC-5 as "a cascade of transducers or modules that, at each step, add structure to the documents and, sometimes, filter relevant information, by means of applying rules". Most of the current IE system follows this general architecture, even though they incorporate their particular system specific components. The following are some of the general information extraction components: Document preprocessing**:** this component allows the IE system to identify language specific characteristics and to normalize the document in order to save computing power, space, and enhancing the system performance. Syntactic analysis**:** partial or full, it helps us to identify the group of phrases like verb, noun, adjective and adverb with their importance levels for further analysis. Domain analysis**:** Focuses on the main features which are extracted to check the relevancy. Template generation**:** finally the extracted information should be prepared for in well format for further analysis, application inputs, reasoning, and performance evaluation. The proposed system architecture of our Event and temporal information extraction (ETIEA) is adopted from the general architecture of an IE system. Our own system specific components are incorporated along with the general architecture. The following figures depict our proposed system architecture.

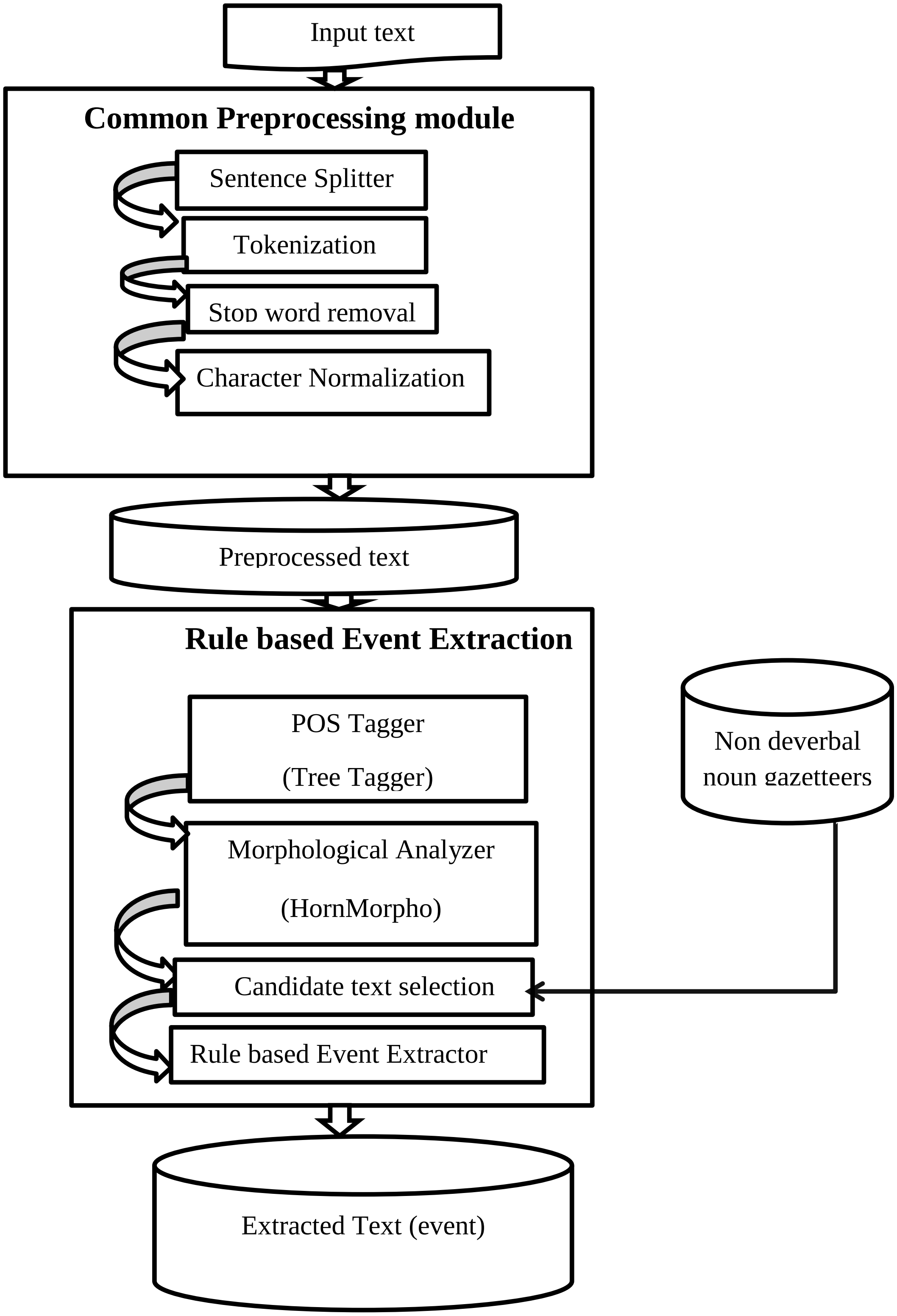


Figure 5-1Model for rule based Event Extraction

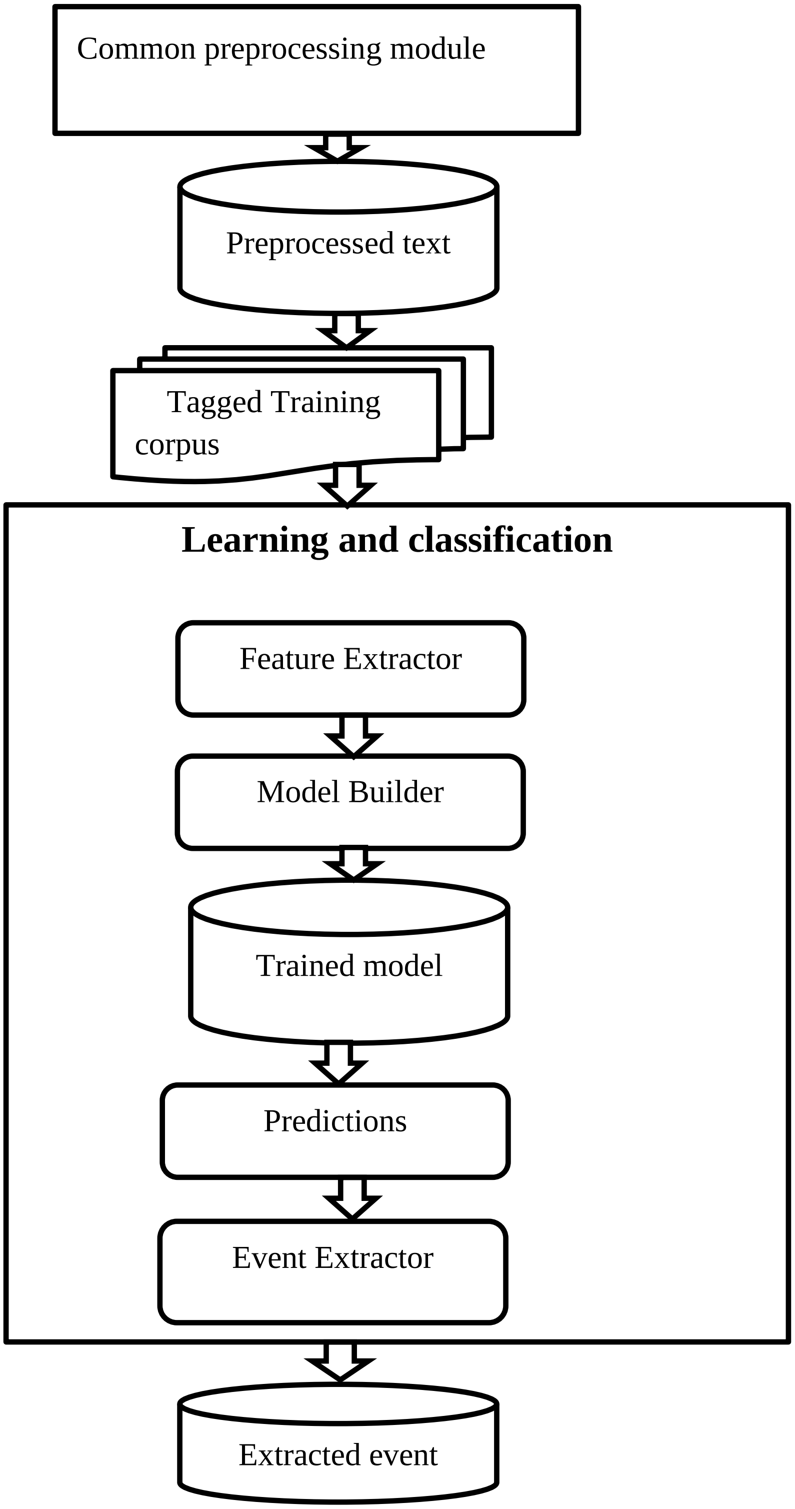


Figure 5-2 Model for machine learning classifier based event extraction

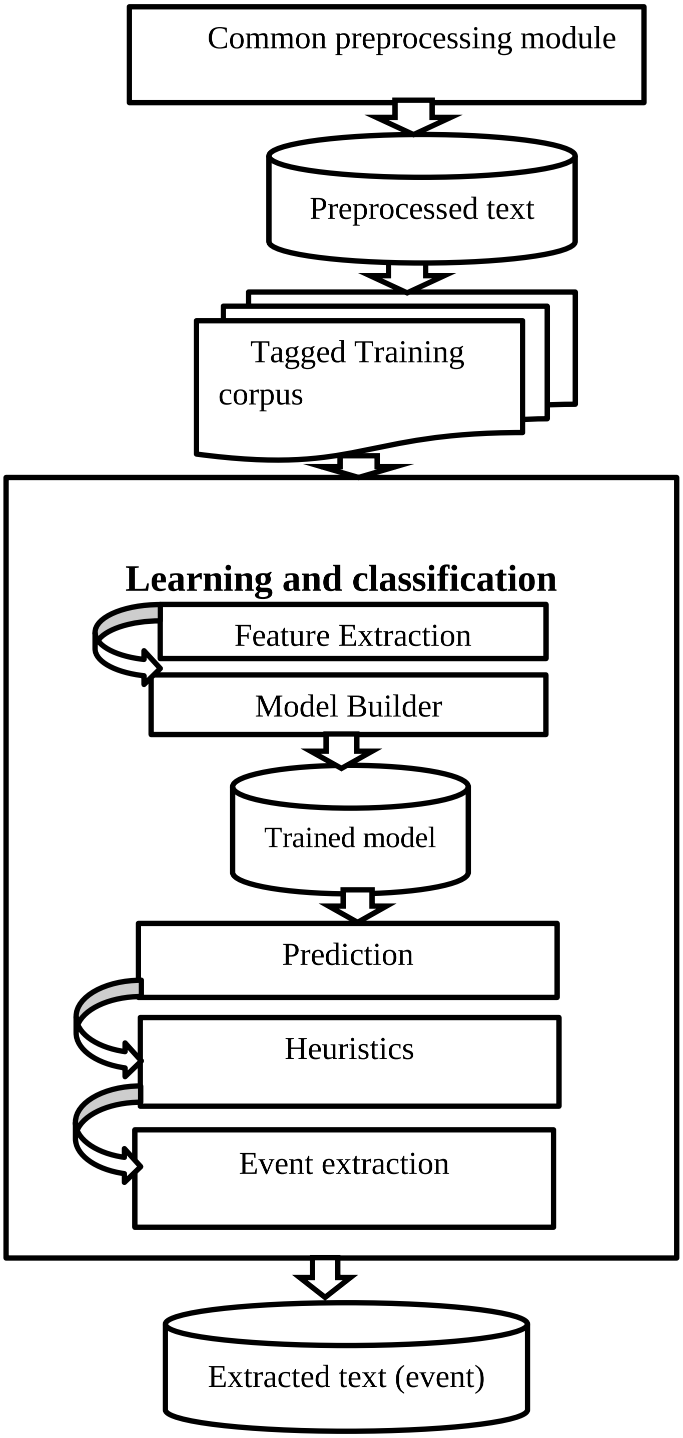


Figure 5-3 Event extraction using hybrid approach model

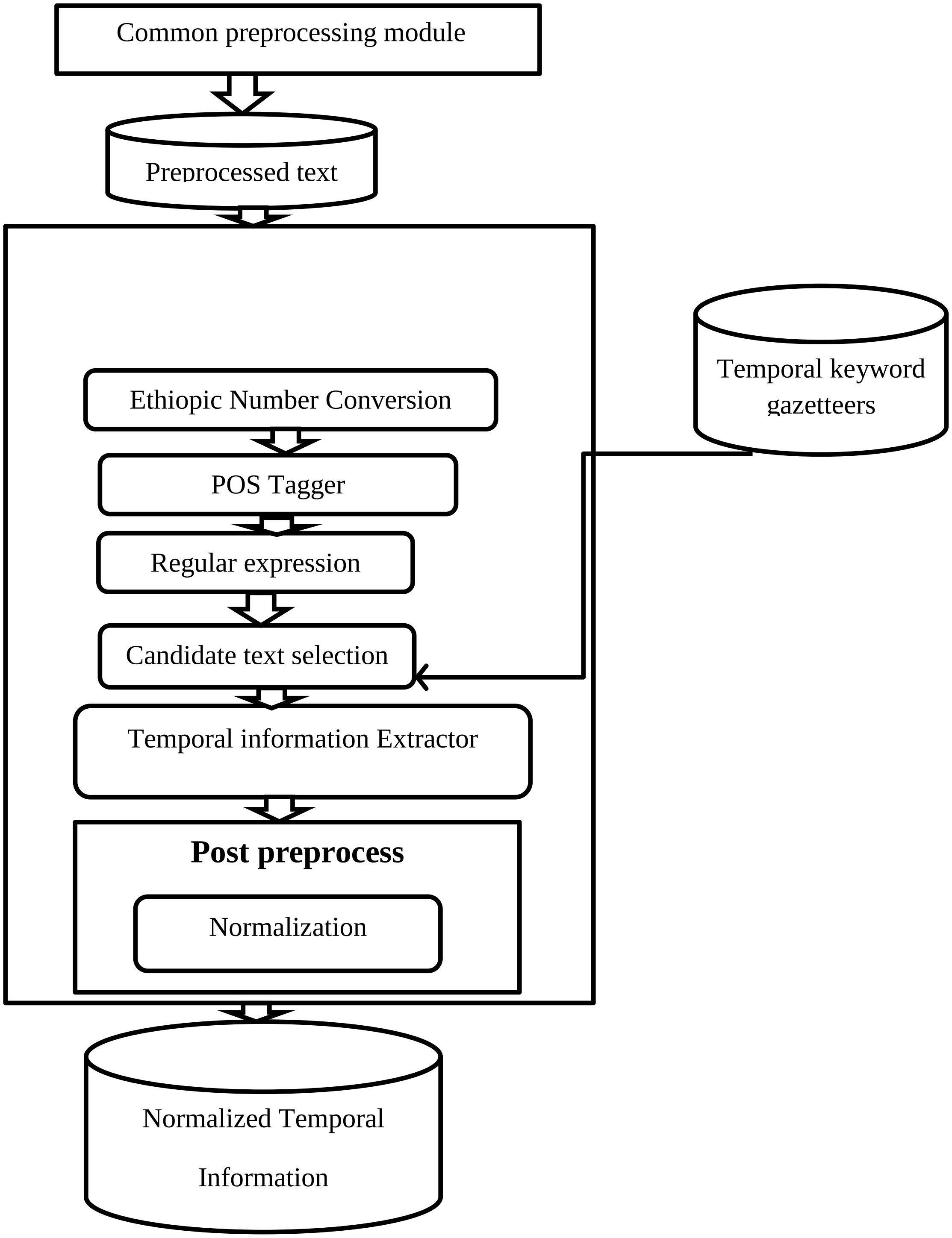

Common preprocessing module
Preprocessed text
Temporal keyword gazetteers
Ethiopic Number Conversion
POS Tagger
Regular expression
Candidate text selection
Temporal information Extractor
Post preprocess
Normalization
Normalized Temporal Information

Figure 5-4 Model for rule based temporal information extraction

5.5. Common preprocessing module

Preprocessing is the most significant and preliminary task of feature engineering approaches in many NLP applications which is used to clean and made ready the corpus for the further processing. The main role of data preprocessing is formatting or normalizing the input documents, so that later tasks can be computed easily. The document preprocessing component handles different language specific issues that are imposed by the nature of the language to make the data ready for remaining phases. In order to get good results, language dependent text preprocessing should be performed before automatic extraction is implemented. Text or document preprocessing is the step by which the text is made comfortable to the learning algorithm or any other component which is in need of the relevant text to proceed to the next level or doing its action accordingly. The preprocessing step is simply a removal of non-informative words or characters from the text in order to save the systems computational resources as well to enhance its performance by getting rid of those unwanted junks. Thus, this Section presents the detail how each module was developed.

Sentence splitter: A sentence splitter divides a spawn of text into sentences. In Amharic a question mark and Amharic full stop (::) are used to end a sentence. Sentence splitting involves the identification of sentences and words of the document to be extracted. The extraction mechanism demands sentence identification since a single sentence could convey a message. Therefore, we have used Amharic full stop (አራት ነጥብ ::), colon and the usual question mark as sentence demarcation.

Tokenization: This is the task of splitting texts in to piece of tokens, which are disjoint and meaning full texts. Sometimes it can be defined as given a sequence of characters and a defined document unit , tokenization is the task of chopping it up into pieces, perhaps at the same time throwing a way certain characters such as punctuations. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing.

In amahric sentence or phrase a single space would be added between the word and punctuations by the system. The tokenizer then tokenizes all the text segments which have space between each other as independent token. Although the selected punctuation marks are used to tokenize the text there is a language specific challenge in Amharic in which, compound words are there such as ቤተ መንግስት፤ ስነ ተዋልዶ. So that we need to have certain rules to keep the consistency of meanings of the compound words for instance by connecting the compound words by hyphen and leaving the symbol ‘-’ during tokenization.

```
    Read input corpus from file line by line
        For each line split them using common
        Amharic sentence demarcation using
        either of (፤, ?, !, ። , )
        Add lines on ArrayList of string
        For each lines in list of lines
          Split tokens using space delimiter
        Return tokens
End for
```

Figure 5-5 Tokenization Algorithm

Character Normalization: As we described in Section four of this document the Amharic language has different characters with the same meaning and pronunciation but with different symbols. The different symbols should be treated equally because there is no change in meaning regardless of the linguistic view of orientation among characters. As described in [62] [ሀ ፤ ኅ ፤ ሐ ፤ሃ፤ኻ፤ ሓ፤ ኃ] [ሰ ፤ ሠ] [ጸ ፤ ፀ][አ ፤ ዓ ፤ ዐ] all these different forms of characters (Fidels) represent the same sound. Therefore, these variants of characters (Fidels) need to be converted to the same or common character form (shape) in order to avoid representing the same words/phrase using different letters having the same sound which will the number of words representing the document without any relevance as a result the performance of the system will down a bit. Such characters are should be normalized to a single character like ሀ ፤ ሰ ፤ ጸ ፤ አ there is no any change in their meaning.

```
get each tokenized words as input
 Store special characters on temporal variable
     Replace them with empty string
Store each characters which has duplicate values
Replace all with one of the representatives
Return words with normalized character/s
```

Figure 5-6 Simple Character Normalizer Algorithm

Stop word removal: like any other language Amharic language has its own list of stop words including conjunctions, articles and prepositions. In our case we adopt stop words used in [47], see in Appendix C of this document. It's obvious that stop words has a great significance to write the meaningful document from linguistic perspective, But when we design an NLP application like IE we need the relevant words to represent the document. As such stop words are frequently occur words in any document without a meaning to describe about the document. So that , in order to enhance the performance of the system as well as to save computing resources we need to get rid of those junks before starting the learning and extraction component.

```
Read Amharic stopwords from file line by line
Sore them on set and add as List
get list of tokens as List
For each set element over the list of tokens
If one of the set element found
discard it
else
```

Figure 5-7 simple Algorithm for Stopwords removal

5.4. Classification Module

5.4.1. Event Detection as classification

Event detection as classification is the task of sequence labeling tokens as on-event and off-event. A Set of used attributes for term classifiers includes: Words or Phrases of each Instance, lemma of each Word and POS Tag of each Word.

5.4.2. Preprocessing the training data

In order to reduce irrelevant information from unstructured plain text, we undertaken necessary preprocessing steps. Since, we used the corpora of walta information center manually tagged corpora of xml file with some unwanted tags such as <? Xml version="1.0"?>, <! DOCTYPE amnews94 (View Source for full doctype...)>, <! -- -->, <amnews94>, <document>, <filename>mes01a1.htm</filename> and <docid> etc are there. To do so, including stop words and special characters we remove those unwanted bags. Next to that by having two tab spaces between tags and the immediate token as special delimiter to tokenize words with their corresponding pos tags. The stem words of the corresponding tokens are generated from the tagger component.

5.4.3. Feature Extraction

Generally, feature extraction for classification can be seen as a search among all possible transformations of the feature set for the best one, which preserves class reparability as much as possible in the space with the lowest possible dimensionality. Features are properties of a text that are used to provide necessary information associated to a given events and increase the confidence level of predicting a token as an event. Feature extractor is responsible for identifying and extracting all the necessary features from the training data. The features we used in this work are directly results from the TreeTagger output, which contains:

- Words of an instance
- POS of the corresponding word
- lemma of the corresponding word

After all the features are extracted and the files are saved as CSV file format. Then using the weka library of converter instances named CSVLoader and ArffSaver we convert to a weka supported arff file format. All the extracted features are supplied to the model builder which will predict the parameters of the model. To classify instances as on-event and of-event, words, pos of the corresponding word, stem of the words with various feature combination are used. Transforming an instance into an array of values, either numeric or nominal, is called feature representation and the array of the values is called the feature vector, which is an input to the classifier.

5.4.4. Model Builder

It is the primary concern of supervised machine learning classifier building a trained model that will be used for the prediction. Building a model is the component designed to estimate the model Coefficients and then build a trained model. Estimation is taken place based on the input from the training corpus that is supplied by extracted features which is generated from feature extractor.

After the features are extracted and the input data is preprocessed to a weka supported file format then it passes through the weka classifiers. The model builder processes starts from reading the arff file format files from the directory. The weka classifier splits the instances using cross validation split as training and testing set. The cross validation split gets input of instances and number of folds which is going to be trained and tested. Then we have a two dimensional array of instances type for n number of folds iterate over them, split the instances for training and testing set using the trainCV and testCv methods of weka instance class methods. We have an array of models we call them the weka classifiers. Now, the classifier accepts an input of selected algorithm, training set splits and testing set splits. To calculate the accuracy the Evaluation classes of weka is used in which we collect every group of predictions for current model in a FastVector. Finally, the model builder builds a model and returns summary of training testing pair using the Evaluation object of to summary string method, class detail string and matrix string methods.

5.4.5. Feature selection

It is the processes of automatic selection of attributes in the training data that are most relevant to the predictive modeling problem. Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method do so by creating new combinations of attributes, whereas feature selection methods include and exclude attributes present in the data without changing them. It can be used to identify and remove irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model. The selection criteria are depending on the probabilities of the respective attribute in the class label values. There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods. But, Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model as used to evaluate a combination of features and assign a score based on model accuracy. The search process is best-first search, it may be stochastic such as a random hill-climbing algorithm, or it may use heuristics, like forward and backward passes to add and remove features.

5.4.6. Prediction phase

It is the processes of predicting class labels for the unseen texts based on the trained model. The task of prediction is classifying terms as off-event and on-event classes. By using the knowledge acquired from the model, detection is performed. The knowledge in model builder contain features that are extracted and stored during the training phase, are supplied to the classifier to identify their respective label. Predictions of classes are performed based on the calculated probability.

5.5. Rule Based Module

Machine learning techniques are powerful and dominant over hand crafted rules because of their learning and dynamic natures. Even though, it has several significances, the corpora development, computation time and accuracy are intensive and costly in comparable with the knowledge acquisition approach. In this work to tackle the loss of performance because of missing to predict most of the nominal events in machine learning approach we tried to use independent rule based approach. Rule-based information extraction has long enjoyed wide adoption throughout industry, though it has remained largely ignored in the academic environment, in favor of machine learning methods. However, rule based systems have several advantages over pure ML systems, including: the rules are interpretable and thus suitable for rapid development and domain transfer; and humans and machines can contribute to the same model.

5.5.1. Event Extraction

We analyzed the machine learning output and the relative percentage of recall is higher than its precision value. Because of that the machine learning algorithm learns pattern and gives equal weight for each candidate in the text to predict their category. So that the machine learning approach loses its advantage of high precision value to mean that the ratio of true positives per the combination of true positives and false positives is lower than that of the ratio of true positives per total number of contents in the text. The other point should be raised is the contingency table of the machine learning result is a little bit ambiguous. This assumption leads as the rule based is powerful than the machine learning approach because of the scope of the hand crafted rule covers is not comparable as that of the machine. The effectiveness of the rules in specific contents it's aimed to catch is totally better than the machine.

To gain this advantage of rule based approach we developed a standalone rule based approach based on syntactic features of words such as pos and morphological analyzer. In addition to this the rules are developed depend on the machine learning algorithms suggestion of best features in which an attribute dominantly helps the classifier to predict the candidates category. To do so, the rule based approach incorporates some components such as Part-of-speech Tagger, Morphological Analyzer, event extractor rules and some non deverbal event gazetteers.

5.5.1.1. POS Tagger

Part of speech tagging is the prominent application in any preliminary NLP systems and linguistic environment, in which it labels words to their part of speech or classes. The task of labeling can be done either using statistical approach or from highly linguistic background through handcrafted rules. Statistical systems can use POS tags as classification features, knowledge based systems as elements of extraction rules. POS tagger could help as a baseline for higher level NLP applications, in which having part of speech of a particular word is crucial for any syntactic or semantic analysis.

In our case we adopt a multilingual decision tree based Tree Tagger. Tree Tagger is a tool for annotating text with part-of-speech and lemma information. It was developed by Helmut Schmid in the TC project at the Institute for Computational Linguistics of the University of Stuttgart. The Tree Tagger has been successfully used to tag various languages including German, English, French, Italian, Dutch, Spanish, Bulgarian, Russian, Greek, Portuguese, Chinese, Swahili, Latin, Estonian and old French texts and is adaptable to other languages if a lexicon and a manually tagged training corpus are available. The training data for tree tagger contains words, POS of the corresponding word with their lemma. We used the data source from walta information center which contains around 204819 tokens with manually annotated tags. We add an automatic construction of lexicons using dictionary of words lemma.

To train the tree tagger we put the tree tagger jar file in our project folder. There are two directories in tree tagger bin and model folders. The bin folder contains tree tagger executable file and generated model file for Amharic. The model contains the amahric parameter files in utf-8 format. First of all we have to change to the home directory of tree tagger then open the command prompt window change to the bin directory then put this command: bin/train-tree-tagger lex.txt unknown.txt train.txt train.par after all the compressed binary files are generated as a model. Here the known text contains out of vocabulary words which has no corresponding lemas. Finally to use the trained model we can use any Amharic plain text the models performance is accurate. We have tested it several times it automatically tags words with their correct classes.

5.5.1.2. Morphological Analyzer

Morphology is the study of word formation–how words are built up from smaller pieces. When we do morphological analysis, then, we're asking questions like, what pieces does this word have? What does each of them mean? How are they combined? From which word it's derived. So that analyzing the words especially for morphologically rich languages like Amharic has a lot of significance. Extraction of morphological information from words which constitute possible word forms lemma, part of speech; other morphological tags depending on the part of speech as such verbs which might have features like tense, aspect, and mod. Words which are ambiguous with respect to certain morphological categories may undertake disambiguation.

In our work events are dominantly recognized as verbs and nouns in a sentence as in [14]. In case of nominal events we get confused to identify events clearly because nouns are not only events they can be person name, location name, organization name etc. Nominal events are also categorized as deverbal and non deverbal in order to disambiguate deverbal nominal events we need a morphological analyzer. We adopted the open source python project: HornMorpho, which is done for Amharic, Tigrgna and Afaan Oromo. HornMorpho analyzes words of the Amharic language into their constituent morphemes (meaningful parts) and generates words, given a root or stem and a representation of the word's grammatical structure. So that, we can easily assure deverbal events, if they are derived from verbs possibly they are events.

5.5.1.3. Non-Deverbal Event Gazetteers

Nominal are categorized into deverbal and non-deverbal. So, events are examined and it can be act as deverbal entities dominantly and sometimes in the form of non-deverbal entities. Since such entities are common and due to their ambiguousness to resolve such issues we construct list of gazetteers. Even though in Amharic there is no any constructed list of those words other than the definition and samples of non-deverbal events we referenced thus non-deverbal entities as in [72]. See in Appendix B of this document.

5.5.1.4. Event Extractor Rules

In the rule based component architecture of ETIEA unstructured text is accepted as input. Pre-processing steps are performed on the input in order to eliminate the stop words from the given text and remove the punctuations and generate tokens at last. For each token POS tags are assigned by using the POS Tagger module. Each and every token in the token that are tagged by the pos tagger are compared with the rules described in the event extractor. If the token is in verb form or action noun it is considered as events and it separated from the other tokens. The problem is the noun tokens are ambiguous we do not have any evidence to say action nouns. In addition to that nominal events can be represented either deverbal nouns or non deverbal nouns. To disambiguate nominal events we adopted the open source morphological analyzer (HornMorpho), because we can easily get the morphological form, root, stem and citation forms of the respective candidate noun tokens. The procedure is repeated with all the other tokens to identify all the events present in the given text.

```
    Read pos tagged text from file
               For each line having two column
           If lines splite with space at second index contains
               list of all verb categories
  Return lines split with space at first index values
   For each token in line of list of string
       If lines splite with space at second index contains list of all noun categories
        If lines split with space at first index contains temporal expression
                  Nothing to do
If lines split with space at first index contains
    Person, place, organization name
        Nothing to do
      Retun lines split with pace at first index values
     End if
End if
```

Figure 5-8 Algorithm for event extractor

```
get list tokens identified and tagged as nouns
      Store those nominal tokens
    Call horn morho anale_fle
Pass as an input for anale_file
   Return an outputfile which contains
morphologically  analyzed of selected words in
unstructured format
Read the hornmorpho output file
   At the begining temporal variable result is empty
If  line is  empty line
   Store each token content add on list of string
Do the result empty
  Nothing to do
Assgin curreunt lines for reslut
For each token in list of strings
   Split token with special deliimeter
Store splited value on an array
If  splited values contains verb citation form or
infinitive form
    Retun the word which has verb citation form and
infinitive form
 End if
End  if
```

Figure 5-9 Algorithm disambiguate nominal events using hornmorpho

5.6. Hybrid Module

In this work we initially we develop a method for event extraction using machine learning. We use a Naïve Bayes, LIBSVM and Decision Tree as the underlying machine learning algorithm. We observe that this machine learning based system often makes the errors in extracting the events denoted by deverbal entities we call them nominal events. This observation prompts us to employ several strategies without the help of machine learning and in conjunction with machine learning. The immediate strategies come to without the help of machine learning approach is the standalone rule based approach, which includes several components such as part-of-speech tagger, morphological analyzer and non-deverbal event gazetteers with some rules. We compare this rule based approach with the machine learning based approach, totally the rule based approach outperform the former by handling deverbal entities. Finally we come up with the hybrid approach for event extraction to see the difference, what if we add some heuristics in conjunction with the machine learning based approach to get the advantage of the machine learning and rule based approaches in combination.

5.6.1. Machine learning component

This component is not new rather it's just the classification module described in Section 5.4. What we did here is having the result of the machine learning classifier i.e predicted instances as on event and off-event classes with their respective values of proportion of the classes for each instance. We store the model then applying some heuristics based on threshold values.

5.6.2. Heuristics

The rule based approach incorporated with the machine learning approach is to tackle the mislabeling of the deverbal entities through the later approach. We seriously analyzed the machine learning approach results and compare with the result of the rule based approach. As we described at the beginning of this Section the machine misses nominal (deverbal) events. The verb triggered events are correctly detected using the machine learning approach. Dominantly the machine learning approach labels those nominal events by giving equal weight for both classes' i.e on-event and off-event. Sometimes and it's almost rear case the machine learning approach labels deverbal entities as off-event classes. Actually nominal events are ambiguous, those can be act as deverbal and non-deverbal. Linguistically, non-verbal word forms are derived from verbal word forms. Various inflectional and derivational morphological rules are involved in the process of evolving from verbal to non-verbal word forms. To do so, we need a special mechanism to tackle this limitation of the machine learning approach. In our case based on our serious analysis we take the threshold values when the machine learning approach gives equal weight for those instances that might be deverbal entities. Our assumption is not without background knowledge of the instances rather the analysis shows in most of the case those instances assigned equal weight by the machine are ambiguous deverbal entities.

The procedures to apply this rule in conjunction with the machine learning approach follow these strategies. First of all store the machine learning result which contains segments with their corresponding class values. As we said earlier we accept positively an instance which has higher on-event class values. Whereas, instances with equal weight values for the corresponding class are the target instances, which are going to be analyzed using heuristics. The rules are simple having those target instances we use the instances morphological behaviors to distinguish deverbal entities. If an instance is nominal derived from verbs using the morphological analyzer dominantly it’s an event. The following is simple algorithm used to apply rules with the machine learning approach assuming the ready machine learning result.

```
get the machine learning result
        For each instances in result
Identify   on_event class instances value
Store them
get off_event class instances values and store on temp
if value of on_event class value for the corresponding instance less than that of the off_event class value
            Nothing to do
else If value of on_event class value for the corresponding instance greater than that of the off_event class value
        Return corresponding instances
else If value of on_event class value for the corresponding instance equals that of the off_event class value
```

```
----Continued
Add instances on set
Nothing to do
Iterate over set
Return each element of the set
Call the anal_file () of hornmorpho
Return the instances morph features
if an instance has verbal derivation or
if an instance has an infinitive form
Add instances on set
else ignore them
Iterate over set
For each instances in set
Check if the set element is found in non-deverbal
Entities gazetteers list
Return unique elements
End for
End if
End if
```

Figure 5-10 Incorporating rules with the machine learning classifier

5.7. Temporal information extraction

Always we define events in terms of point of time, duration and frequency. Temporal information is a crucial source of information about when something happens, how often something occurs or how long something lasts. To do so, automatic extraction of temporal expression in a text is crucial for event relation and ordering tasks. In our case we develop simple regular expression, temporal keyword gazetteer list, syntactic features and context of words based method to extract temporal expression from Amharic unstructured text.

5.7.1. Temporal information extractor

A temporal expression is some sequence of words or may be numerals that tell us when something happened, how long something lasted, or for how often something occurs. Temporal expressions may be calendar dates, times of day, or durations, such as periods of hours, days, or even centuries [63]. Extracting temporal information from text helps as to answer when or for how long an event is happened in certain situation. In some aspects of life we need chronological ordering of events and the cause and effect relationship between events. So this may have brief insight through the sequence of happening of an event within a particular time frame. Our focus in this work is to extract temporal information without any relation and order construction. To do so, since temporal information indicators such as time adverbs as well as date time expression are not as much content as other textual information in any text. We choose a powerful rule based approach and regular expressions. We are not deciding to use this approach without evidence, it's because of as in [37] the state-of-art for timex extraction is publically available tool we call it HeidelTime temporal tagger. HeidelTime works based on rules, patterns and resources we don't need to go detail explanation of it, but the point is with a convenience rule based approach scores better performance over other approaches for this particular task.

5.7.2. POS Tagger

We described the POS tagger we used in Section 5.5.1.1 of this document for event extraction task. In this task i.e for event extraction component we employed POS tagger to reduce the set of interest among the target groups to detect temporal expressions. In our case the tags NUMP and NUMCR, which refers cardinals and cardinals with preposition are the targets which could be analyzed. We get the content which is tagged as these possible tag set then we applied a set of rules again to check whether the candidates are valid temporal expressions.

5.7.3. Regular Expression

Regular expressions are the most useful tools and formal language for specifying string. Regular expressions are placed inside the pair of matching. A regular expression, or RE, describes strings of characters (words or phrases or any arbitrary text) [58]. It's a pattern that matches certain strings and doesn't match others. A regular expression is a set of characters that specify a pattern. Regular expressions have served as the dominant workhorse of practical information extraction for several years. However, there has been little work on reducing the manual effort involved in building high-quality, complex regular expressions for information extraction tasks [56]. Large class of entity extraction can be performed using complex regular expression construction. Among the entities which are amenable for the regular expression email address, credit card numbers and date time expressions are belongs to this group.

In this work we developed regular expression based methods temporal information extraction in amahric unstructured text. The method utilizes handcrafted rules and regular expressions to recognize temporal expressions from the input text. As we described in Section 4.7 of this document temporal information in the input text might include date-time expressions, duration or periods, and frequency. We build a dictionary of temporal trigger keywords like temporal adverbs. Actually in any language temporal expressions are too small than other information, so that it's adequate to extract them developing simple handcrafted rules in combination with other syntactic features. The context of words from their neighbors is also important information which helps us to extract temporal information. If a word in the input text is preceded and/or followed by certain temporal trigger keywords or date time expressions most probably the word is also temporal expression. For example if we take the phrase "መስከረም 12 ቀን 2009 ዓ.ም" we match the year 2009 using a regular expression for valid year between 1900 to 2099 (19|20)[0-9][0-9]) now the token which is found before it or the (n-1) term is temporal keyword.

Again the (n-2 and n-3) term is a temporal expression. Here to find the previous four terms we have list of temporal gazetteers for the term “ቀን”and list of month names also day of week having this matching the regualr expression to check the valid year, month, and date format by applying such conditions our system recognize this phrase is a temporal expression. So, the token sequence is important in recognizing the temporal expressions. We also developed simple regular expression to check if the input text contains valid (year, month, and date) or (month, date and year) or (date, month and year) format respectively with either of hyphen or forward slash or dot delimited.

The other point is our calendar is 13 months in which the last month is special as leap years, with probably five or six days, whereas the year is always similar. In Amharic text to detect the valid date and month “ጳግሜ” we used this simple regular expression (0[1-6]|[1-6])[- /.](1[3]). This regular expression checks the date is either from 1 up to 6 or 01 up to 06 whereas the month is 13. The following table contains some of frequently used special characters with their meaning in regular expression.

Table 5-1 regular expression meaning

Characters	Regular Expression meaning
.	Any character, including whitespace or numeric
?	Zero or one of the preceding Character
*	Zero or more of the preceding character
+	One or more of the preceding character
^	Negation or complement

Regix for time pattern "([0-9]|0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]"

Regix for year, month, date format separated by either of the three delimiters "(19|20)[0-9][0-9][- /.](0[1-9]||[1-9]|1[012])[- /.](0[1-9]||[1-9]||[12][0-9]|3[01])"

Regix for month, date, year format separated by either of the three delimiters "(0[1-9]||[1-9]|1[012])[- /.](0[1-9]||[1-9]||[12][0-9]|3[01])[- /.](19|20)[0-9][0-9]"

Regix for date, month, year format separated by either of the three delimiters "(0[1-9]||[1-9]||[12][0-9]|3[01])[- /.](0[1-9]||[1-9]|1[012])[- /.](19|20)[0-9][0-9]"

Regix for year “(18|19|20)[0-9][0-9]"

Regix for date, month/ጳግሜ, year format separated by either of the three delimiters "(0[1-6]||[1-6])[- /.](1[3])[- /.](19|20)[0-9][0-9]"

Regix for year, month/ጳግሜ, date format separated by either of the three delimiters "(19|20)[0-9][0-9][- /.](1[3])[- /.](0[1-6]||[1-6])";

Regix for month/ጳግሜ, date, year format separated by either of the three delimiters "(1[3])[- /.](0[1-6]||[1-6])[- /.](19|20)[0-9][0-9]";

Figure 5-11 Regular expession to extract date time patterns

5.7.4. Candidate text selection

Temporal information in a text is well known and is not too much in numbers as other lexical information. The sensitive key words used as indicator are month names, day of week, temporal adverbs, duration, set, frequency, date in numerals and time. In this work we prepared a list of temporal trigger keyword gazetteers. The next activity is identifying those possible candidate texts from the unstructured text using those incorporated gazetteers and other syntactic features including pos and complex regular expressions.

5.7.5. Temporal information normalization

The temporal expression normalization task is an interesting and challenging one [60] because, while some temporal references appear in well-defined formats, others are expressed using a wide range of natural language constructions, and are often ambiguous, requiring analysis of the surrounding text in order to arrive at an interpretation.

Date Time expression Normalization: Text normalization, in general, aims to map an excerpt of text to a standard form, such that the output is more amenable to further processing or interpretation. A temporal expression is a word, phrase, or clause that in some way evokes a period of time. Many temporal expression normalization schemas facilitate the representation of sets of calendar dates using single strings. For example, single dates might be represented in date-string form **YYYY-MM-DD,** where YYYY is a year, MM is a month, and DD is a date. The temporal information indicator expression in Amharic language is defined in variety of forms including time adverbs, Date and time expressions. So that, temporal expressions should be normalized as such either of a particular date, or interval of dates, or a single day or interval of a definite quantity of two or more days not explicitly anchored to a particular region of the calendar. For instance , The former Ethiopian Prime minister Mr. Meles Zenawi has died in August, 14 , 2004 ዓ ም/ የቀድሞው የኢትዮጵያ ጠቅላይ ሚኒስተር አቶ መለስ ዜናዊ ንሃሴ 14 ፣ 2004 ዓ ም ከዚህ አለም በሞት ተለዩ። In this sentence the time expression should be normalized to 2004-12-14. There are also expressions which are directly anchored by the calendars in which certain natural classes of date intervals are commonly referred to in text and are easily represented by assigning variable values to some of the placeholders in the date-string form. A key property of such time periods is that their start and end dates can be easily read from the calendar. For example represents March of 2003 /መጋቢት 2003, or [2003-07].

5.7.5.1. Ethiopic number conversion

Ethiopic is the term used to refer primary writing system of Eritrea and Ethiopia. Other terms that have been used for the script in the west have been Abyssinian, Ethiopian and Abyssinic [65]. In Eritrea and Ethiopia the writing system is known affectionately as Ge'ez, Fidel, and Fidelat. The Ethiopian numeral system was devised following the ancient Greek Method of modifying members from the character set for the spoken language. The characteristics over bars and under bars indicate a likely Roman influence as well. Unlike the Greeks or the Romans the Ethiopic numerals show a greater mutation from the spoken letters that they may have been based upon.[64] Also, the arrangement of the numerals in the formation of numbers doesn't have the same kind of cyclic behaviors found in Roman, Greek OR Arabic stems. The algorithm for the glyph arrangement is not immediately apparent so that it's difficult to understand even for the native users. The glyphs used to construct the Ethiopic numeral system are as follows:

Table 5-2 Ethiopic Numeral System

Ones	Arabic	0	1	2	3	4	5	6	7	8	9
	Ethiopic		፩	፪	፫	፬	፭	፮	፯	፰	፱
Tens	Arabic	10	20	30	40	50	60	70	80	90	
	Ethiopic	፲	፳	፴	፵	፶	፷	፸	፹	፺	
Higher	Arabic	100	1000	10,000							
	Ethiopic	፻	፲፻	፼							

So, in Amharic unstructured text temporal information sometimes might be represented either in the form of Ethiopic Numeral system or arabic nuber system. But, frequently arabic number system is used. Basically, Arabic number system is as easy as in comparison with the Ethiopic numeral system in which we can easily catch them using either regular expression or sequence of rules . In our work we used regular expressions and rules to detect and extract valid date time expressions in order to make it simple for the regular expression usage we convert the Ethiopic numerals to Arabic number system. Now, the problem is once we get such Ethiopic Numeral system in Amharic text, that might represent temporal information, how to tackle them. To do so, we develop an algorithm to convert the Ethiopic numeral system to the Arabic Number system. Actually, the conversion is really tedious task in which converting from Arabic number system to Ethiopic number system is not difficult as converting to the reverse. In conversion process we used the usual hierarchical mathematical notations.

```
get input string from input text

assign constant values of unicode for ones ,ten, hundred, thousand and ten thousand digits of Ethiopic character

check whether the character is Ethiopic number using

if ones value  is less than the character or the character is less than the ten thousands unicode value return true

confirm that we have a valid input string using

if input String is null or its length is zero return failure message

iterate over input string length

if char at certain index is not Ethiopic number

return failure message

end for

end  if

read right to left to get the orders of 10

assign values for temporal variables outputNumber=0, power=1, lastPlace=0 and firstNumber is input string length minus 1

iterate over firstNumber

assign values of char at certain index temporal variable ethNumber

if ethNumber is equal to hundred or ten Thousand

   if lastPlace is equal to 1

power*=10

end if

else if lastPlace is greater than or equal to three

the previous number is either ፻ or ፼
```

```
In case if ፻ or ፼ are the last number on the left
If the number is equal to firstNumber
Power=100
End if
if ethNumber is equal to ten thousand
                power *= 100
in case of single number
   if  input number Length is equal to 1
               assign  power for output number
else if number is equal to zero or input number Length greater than one
add power on output number
   if ethNumber is equal to hundred
 lastPlace is three else
 lastPlace is four
        if ethNumber is greater than ten
           ethNumber is its value minus nine
           if  number is equal to firstNumber or lastPlace is not equal to one
 power *= 10
  lastPlace is two
else
 lastPlace is one
        ethNumber is its value minus Ounces digit minus one
        outputNumber  will be ethNumber * power
        power *= 10;end for
return outputNumber
```

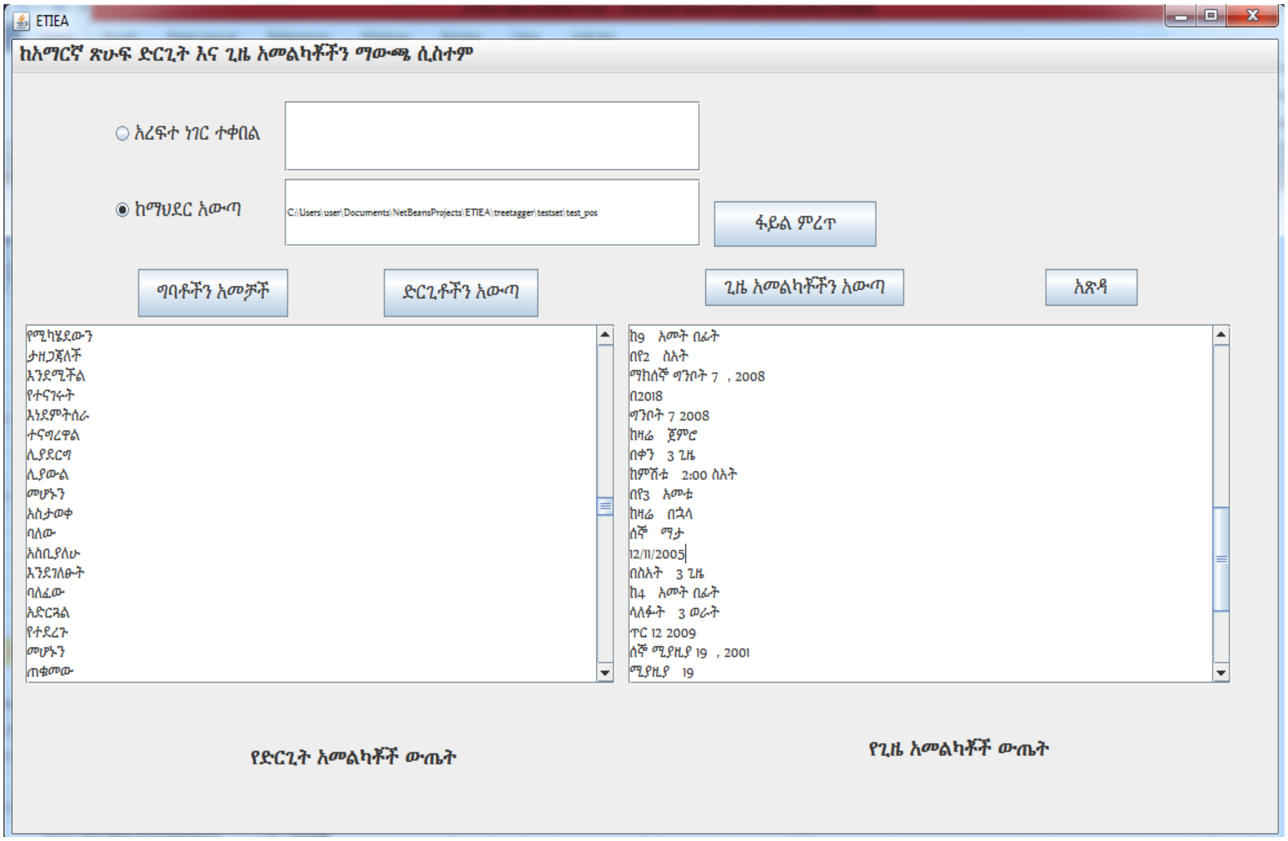
Figure 5-12 Ethiopic Number conversion Algorithm

Table 5-3Conversion of normalized temporal expressions

Types	Example	Normalized
Dates	2008-01-01	2008-01-01
Dates	ሰኞ መስከረም ፲፪ ፲፱፻፹፰	01-12-1988
Date time	ሐሙስ መስከረም 18 ቀን 2009 ዓ.ም	01-18-2009 ዓ.ም
Months	2008-02	2008-02
Date time	05/13/2009	05/13/2009
Year	፲፱፻፶፭	1955
Time	2:00	2:00
Date time	መስከረም 12 ቀን 2006	
Date	መስከረም ፳፩	መስከረም 21
Date	ሚያዚያ 19 2001	
Date	ሚያዚያ 19	
Date	ሰኞ ሚያዚያ 19 , 2001	08-19-2001
Date	ጥር 12 2009	05-12-2009
Duration	ላለፉት 3 ወራት	
Period	ከ4 አመት በፊት	
Set	በቀን 3 ጊዜ	
Date	30-12-1998	30-12-1998
Date	ሰኞ ማታ	
Time	ከምሽቱ 2:00 ስአት	ከምሽቱ 2:00
Date	05/13/2009	05-13-2009
Set	በየ3 አሙቱ	
Duration	ትላንት ሌሊቱን ሙሉ	
Period	ከዛሬ በኋላ	
Date time	ከጥቅምት 15 2008 ጀምሮ	02-15-2008 ጀምሮ
Date	መስከረም 20፣ 2001	01-20-2001

5.8. User interface prototype

Based on the proposed designed model we developed a prototype to evaluate the performance of the system. The extracted data from Amharic unstructured texts are then stored in the file for further use by users or any other application. In the figure below we tried to show the snippet of the user interface prototype which contains the results of ETIEA.



C:\Users\user\Documents\NetBeansProjects\ETIEA\treetagger\testset\test_pos

```
የሚካሄደውን
ታዘጋጃለች
እንደሚችል
የተናገሩት
እነደምትሰራ
ተናግረዋል
ሊያደርግ
ሊያውል
መሆኑን
አስታወቀ
ባለው
አስቢያለሁ
እንደገለፁት
ባለፈው
አድርጓል
የተደረጉ
መሆኑን
ጠቁመው
```

```
ከ9   አመት በፊት
በየ2   ስአት
ማከሰኞ ግንቦት 7  , 2008
በ2018
ግንቦት 7 2008
ከዛሬ   ጀምሮ
በቀን   3 ጊዜ
ከምሽቱ   2:00 ስአት
በየ3   አመቱ
ከዛሬ   በኋላ
ሰኞ   ማታ
12/11/2005
በስአት   3 ጊዜ
ከ4   አመት በፊት
ላለፉት   3 ወራት
ጥር 12 2009
ሰኞ ሚያዚያ 19  , 2001
ሚያዚያ   19
```

Figure 5-13 sample user interface prototype for ETIEA

For the development of the prototype ATIE system Java programming language is used. We also employed tools including TreeTagger, Hornmorpho, Jython library and Weka open source machine learning algorithms. The reason we choose the java programming language is that machine learning algorithms of Weka are developed using java which can easily be imported and used in the java environment. The publically available language independent part-of-speech tagger, which is TreeTagger, is used to annotate Amharic texts with their proper part-of-speech. Whereas the Jython library is used to integrate the python based morphological analyzer for Amharic to get words morphological features. The other tools we used are an open source data mining tool weka which contains machine learning classifier algorithms to classify instances as on-event and off-event.

To use the prototype the Amharic sentence is entered as input if we select the radio button “ዓረፍተ ነገር ተቀበል” or Amharic text is opend from file where it is found if we select the radio button “ከማህደር አውጣ” using the “ፋይል ምረጥ” button. After the text is entered or opend first it will be preproccessed by the “ግባቶችን አመቻች” button, which splite sentences/ tokenize , remove stop words, normalized and tagged with their part-of-speech etc then it will be saved on file. After the input text preproccessed events are extracted using “ድርጊቶችን አውጣ” button ,which extract verb triggered events, disambiguate and extract nominal events, incorporate and check non deverbal events, check context of words with their corrosponding tasggs.Temporal information is extracted using “ጊዜ አመልካቾችን አውጣ” button the different subcomponents are called and applied to extract temporal information. The other two components for event extraction is using machine learning classifier and hybrid approach which is included in the protype in jemenuItem1 and jemenuitem2. When the “የማሽን መማርን በመጠቀም ድርጊቶችን አውጣ” menu item pressed or (Ctrl+M) the result of the machine learning classifier result is displayed with extracted events. When the “ድቅል ዘዴን በመጠቀም ድርጊቶችን አውጣ” menu item pressesd or hold(Ctrl+Alt+H) key the hybrid approach component result is displayed. In summary, the ETIEA prototype contains components and subcomponents for event and temporal information extraction. Mainly the components are event extraction using machine learning based classifier , rule based approach, and hybrid approach as well for temporal information extraction using rule based approach are presented with their subcomponents.

CHAPTER SIX

6. EXPERIMENT

This chapter is all about the experimental cases along with their procedures. The other issue addressed in this chapter is performance evaluation including result and discussion for each experiment with different scenarios under the two tasks.

6.1. Experimental procedures

To assure how our work meets the design goals we conducted different experiments for each model. In the following Section the evaluation metrics along with the corresponding results of the proposed ETIEA system will be presented. The standard methods that are used to evaluate a classifier model, rule based model and the hybrid model are used to evaluate the performance of our system. We follow necessary procedures for each approach to conduct the experiment from the beginning to the end.

6.1.1. Data collection

As we tried to describe in Section 5.2 of this docuemnt our data source is Amharic corpus which is prepared by the Ethiopian Languages Research Center of Addis Ababa University in a project called "The Annotation of Amharic News Documents” [57]. In our system we used different training and testing sets for each model based on the necessities of the data required for the corresponding model.

6.2. Performance evaluation

Since, Event and temporal information extraction for Amharic text (ETIEA) is one task of information extraction. As any information extraction system we used standard metrics to evaluate the performance of our system. In these work different components with different approaches evaluated using appropriate test set and metrics. As described earlier in Section 3 of this document the metrics used to measure the performance of our work are the standard precision, recall and F-score.

6.2.1. Test data set preparation

In this study we applied different methods for the event extraction tasks. The machine learning classifier based event detection is intensive and takes lot of time to prepare the training and testing data set. Actually we used the weka open source tool which contains different machine learning algorithm implementation. Weka environment support different model evaluation techniques four different evaluation techniques such as training data set, using supplied test set, using n fold cross validation and using percentage split. The training set option prepares a model on the entire training dataset, and then evaluates the model on the same dataset. So the classifier memorizing the training data pattern and achieve perfect score which would be misleading. The supplied test set split the dataset manually using another program. Prepare the model on the entire training dataset and use the independent test set to evaluate the performance of the model. This is a good approach if there is a large dataset (many tens of thousands of instances). In case of cross validation it split the dataset into k-partitions or folds of equal size. Train a model on all of the partitions except one that is held out as the test set, then repeat this process creating k-different models and give each fold a chance of being held out as the test set. Then calculate the average performance of all k models. This is the gold standard for evaluating model performance, but has the cost of creating many more models. Whereas the percentage split randomly split the dataset into training and a testing partition each time evaluating a model. This can give a very quick estimate of performance and like using a supplied test set, is preferable only when there is a large dataset. Among the above different evaluation options in Weka we use the 10 fold stratified cross validation technique for the instance classification and IE as it is good to get a better result out of the classifier. Therefore the test data set are automatically used from in proportion from the training data set using the cross validation technique.

6.2.2. Evaluation of Event Extraction Component Using Machine Learning Classifier

Event extraction task is designed to extract events from unstructured Amharic text. For these particular component different approaches used such as machine learning classifier, rule based and hybrid approach. Under machine learning approach among the different machine learning classifier algorithms LIBSVM, Decision tree and Naïve Bayes classifier are used for evaluating their performance. Even though we selected best features using weka subseteval attribute selection methods. In order to see the effect of a combination of different attributes in the prediction of the category process, we considered three different scenarios.

Scenario 1: Using all features

Scenario 2: Using all features except stem of the word

Scenario 3: Using only the pos feature

For all the above scenarios the Naïve Bayes, Decision Tree and LIBSVM algorithms are used for the experiment. The experimental result of the algorithm which outperforms than others in each scenario is presented and well explained. The testing option used is 10 fold stratified cross validation.

Scenario 1:

Among the three algorithms selected for experiment the Naïve Bayes classifier performs better than the other classifiers by correctly classifying 90.11% of the instances correctly, while the J48 decision tree follows by correctly classifying 89.09 and LIBSVM by performing 76.76%. All the instances that are used for training and testing the IE component are 10,037. From the total number of instances the number of correctly classified instances and incorrectly classified instances are described as summary result. The other two important histories about the classier result are confusion matrix and detailed accuracy by class. The confusion matrix shows summary of true positive and false positives with the level of confusion the classifier wrongly classified instances. Detailed accuracy by class of the classifier summary includes TP rate, FP rate, precision, recall, F-measure and ROC area for each class and the weighted average of them. TP Rate indicates the rate at which the classifier correctly predicts the class of the token. The ROC Area indicates the curves in identifying the true positive over the false positive instances. The detailed experimental result of Naïve Bayes algorithm is presented below.

Experimental Summary Results

```
======================

Correctly Classified Instances        460             91.6335 %

Incorrectly Classified Instances       42              8.3665
```

Table 6-1 Detailed Accuracy By class for scenario 1 using Naive Bayes Classifier

TP Rate	FP Rate	Precision	Recall	F-measure	Roc-area	classes
0.957	0.202	0.932	0.957	0.944	0.94	OFF-EVENT
0.798	0.043	0.866	0.798	0.831	0.94	ON-EVENT
0.916	0.161	0.915	0.916	0.915	0.94	Weighted Av.

Table 6-2 confusion matrix for scenario 1 using Naive Bayes classifier

OFF-EVENT	ON-EVENT	CLASSES
357	16	**OFF-EVENT**
26	103	**ON-EVENT**

Scenario 2:

Among the three algorithms selected for experiment the Naïve Bayes classifier performs better than the other classifiers by correctly classifying 89.49% of the instances, while the J48 decision tree follows by correctly classifying 89.07% and LIBSVM by performing 76.72%. The following table shows detailed experimental result of Naïve Bayes algorithm for scenario two.

Experimental Summary Results

```
======================

Correctly Classified Instances        454             90.4382 %

Incorrectly Classified Instances       48              9.5618 %
```

Table 6-3 Detailed Accuracy By class for scenario 2 using Naive Bayes Classifier

TP Rate	**FP Rate**	**Precision**	**Recall**	**F-measure**	**Roc-area**	**classes**
0.971	0.287	0.907	0.971	0.938	0.937	OFF-EVENT
0.713	0.029	0.893	0.713	0.793	0.937	ON-EVENT
0.904	0.221	0.904	0.904	0.901	0.937	Weighted Av.

Table 6-4 confusion matrix for scenario 2 using Naive Bayes classifier

OFF-EVENT	**ON-EVENT**	**CLASSES**
362	11	**OFF-EVENT**
37	92	**ON-EVENT**

Scenario 3:

In this case we need to evaluate the performance of the selected attribute which mainly helps the classifier to predict the class of the event without any combination with the other features. Among the three algorithms selected for experiment the Naïve Bayes classifier, LIBSVM and J48 decision tree algorithms performs equal by correctly classifying 89.09% of the instances. The following table shows detailed experimental result of Naïve Bayes algorithm for scenario two. The following table shows detailed experimental description of LIBSVM.

Experimental Summary Results

```
======================

Correctly Classified Instances        452              90.0398 %

Incorrectly Classified Instances       50               9.9602 %
```

Table 6-5 Detailed Accuracy By class for scenario 3 using LIBSVM Classifier

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.971	0.302	0.903	0.971	0.935	0.834	OFF-EVENT
0.698	0.029	0.891	0.698	0.783	0.834	ON-EVENT
0.9	0.232	0.9	0.9	0.896	0.834	Weighted Avg.

Table 6-6 confusion matrix for scenario 3 using LIBSVM classifier

OFF-EVENT	ON-EVENT	CLASSES
362	11	**OFF-EVENT**
39	90	**ON-EVENT**

6.1.1. Evaluation of Event Eextraction using Rrule Based Aapproach

We get a promising result using a machine learning classifier for extraction task. The problem is resided on deverbal entities ambiguousness. Our standalone rule based approach is designed to tackle the limitation of the machine learning classifier and results better accuracy than the former approach. Mainly our focus is on deverbal entities which act as events, but mislabeled through the machine learning classifier. Next to this due to linguistic features verb trigged events are also sometimes get equal weight by the classifier with that of non-event class. This is also another reason which motivates us to come up with the new method in order to get rid of such kind of confusion. In this method in order to make simple analysis and comparison with the other methods we used similar data set. As we discussed in Section 5.5 this module includes some major components such as common preprocessing processes, pos tagging process, morphological analyzer, event extractor rules and we incorporated list of non deverbal entities gazetteer. We used a total numbers of 2536 tokens among these true positive events are 907, false positive events are 22, false negative events are 42, and true negative events are 1565. To elaborate the contingency table values, true positive events are correctly extracted event and exactly they are an event. False positive events are false alarms in which actually they are an event, but they are missed and act as non-event tokens. The false negative value refers those tokens in real they are an event but they are missed and assigned as non-event. Whereas the true negative values are exactly they are not an event also they are assigned as non-event by the system. The following contingency table refers to summarize those values obtained using the rule based approach.

Table 6-7ontingency table for rule based approach performance evaluation

	Event(positive)	Non-event(negative)
Event(positive)	907	22
Non-event(negative)	42	1565

From the contingency table we notice that the level of confusion is less that of the machine learning classifier. To evaluate the performance of this particular approach we need to calculate the usual precision, recall and f-score using their standard formulas described in Section 2 of this document

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

=907/ (22+907)*100

=0.97*100

= **<u>97.6%</u>**

Now, recall is calculated as follows:

$$recall = \frac{true\ positives}{true\ positives + false\ negative}$$

=907/(42+907)*100

=0.955*100 = **<u>95.5%</u>**

Finally Fl-score is the weighted average of the precision and recall values and it can be calculated as follows:

$$F - score = \frac{2 * \ p * r}{p + r}$$

= (2*95.5*97.6)/ (97.6+95.5) = **<u>96.54%</u>**

Table 6-8 summary of rule based event extraction component

Approach	Precision	Recall	F-Measure
Rule based event extraction	97.6%	95.5%	96.54%

As scholars definition precision is the measurement of systems accuracy, whereas recall is the measurement of completeness. From the above result, we notice that the precision value is greater than that of the recall value unlike the machine learning classifier. This result is actually better than the overall machine learning classifier based methods. This is because of that in this particular method we employed the morphological analyzer and other syntactic features of words to disambiguate the deverbal entities. We assert that our assumption, designs meet the target and it results better accuracy with a significant difference in comparison with the former approach for the similar event extraction task.

6.1.2. Evaluation of Event Eextraction using Hhybrid Aapproach

The third and last approach used for event extraction in this work is using hybrid approach by incorporating some heuristics with the machine learning classifier result. We described the reason why we need to employ this techinique for event extraction as extension to the previous two popular techniques in Section 5.6. The performance of this method relay on the power of having the advantage of the rule based and machine learning based methods in conjunction. From the literature background scholars argue that hybrid approach is more powerful than the other two approaches, even though it's intensive to do it. As we said earlier to make a fair judgment on each technique performance we used the same data source. The machine learning classifier label instances as on-event and off-event binary classes by assigning different weights. Instances get highest probability to be an event is the one which is categorized as on-event class by the classifier. In the other case the off-event class instances are mostly non-event. Having these two important issues we are eager to see what if we combine these two independent methods. We accept positive predicated values as its i.e instances categorized as on-event with high weight value, because it's predicted as event, while instances getting equal weight by the classifier in both class are going to be the target instances for the heuristics. In order to get the false negative and the false positive we used a manual scanning of the text to be accurate. This technique in conjunction results 965 true positives, 20 false positives, 38 false negatives and total number of 1470 true negatives. From these results the heuristics contributes 9.7% of the total extracted instances which are ignored by the classifier. The following contingency table refers to summarize those values obtained using the rule based approach.

Table 6-9 contingency table for hybrid event extraction performance evaluation

	Event(positive)	Non-event(negative)
Event(positive)	965	20
Non-event(negative)	38	1470

From the contingency table we notice that the level of confusion is not exaggerated in comparison with the machine learning classifier as well as the rule based approach. To evaluate the performance of this particular approach we need to calculate the usual precision, recall and f-score using their standard formulas described in Section 2 of this document

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

=965/ (20+965)*100

=0.977*100

= **97.9%**

Now, recall is calculated as follows:

$$Recall = \frac{true\ positives}{true\ positives + false\ negative}$$

=965/ (37+965)*100

=0.96*100

= **96.3%**

Finally Fl-score is the weighted average of the precision and recall values and it and calculated as follows:

$$F - score = \frac{2 * \ p * r}{p + r}$$

= (2*96.3*97.9)/ (97.9+96.3)

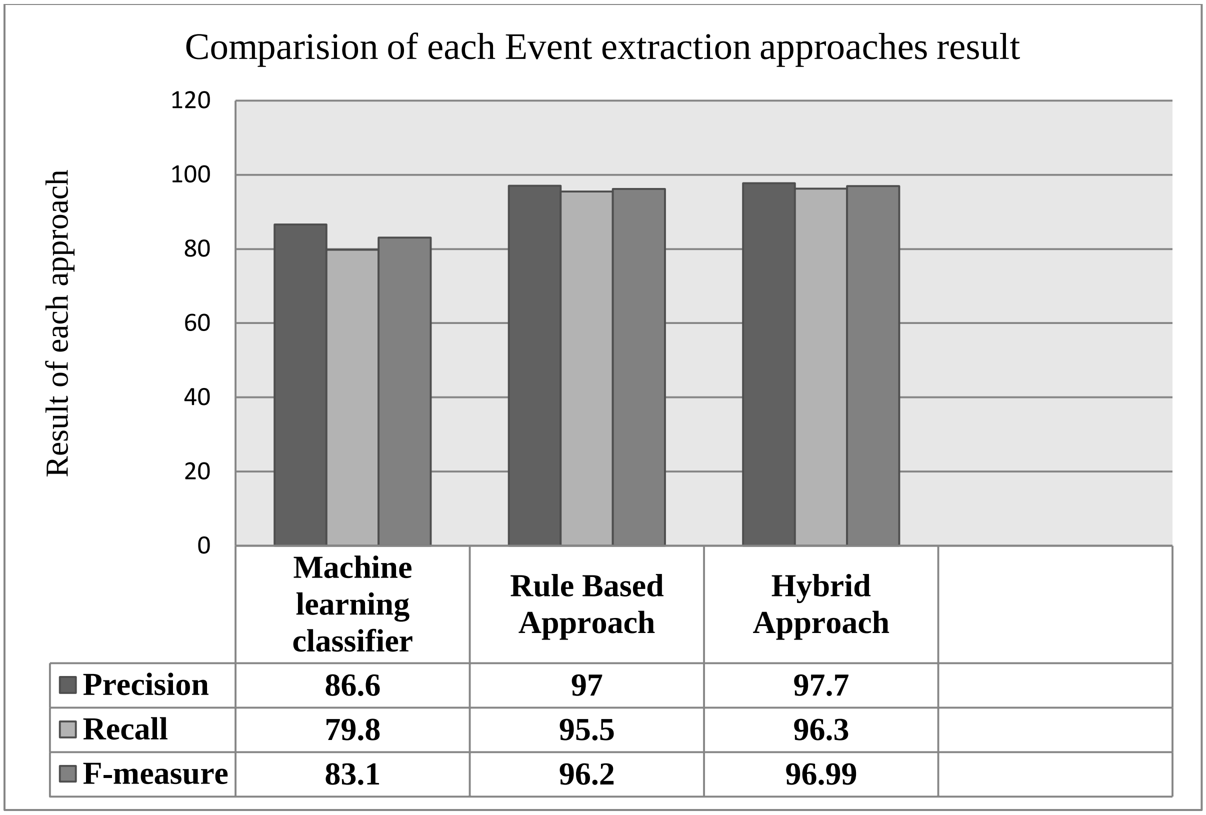
=**97.01%**

Table 6-10 summary of hybrid event extraction performance evaluation

Approach	**Precision**	**Recall**	**F-Measure**
Hybrid approach for event extraction	97.9%	96.3%	97.01%

6.1.3. Comparison of Event extraction Approaches performance

In this work we developed three different approaches for event extraction component. Initially, we developed a supervised machine learning classifier based event extraction system. But, this approach suffers in detecting deverbal entities, which is anonymously act as events. To resolve this problem we propose another rule based approach. This approach is based on syntactic features of words such as pos and context of words as well as morphological features of words to disambiguate deverbal entities. List of non deverbal entities, which are actually events are incorporated as gazetteers. Thereafter we came up with hybrid approach to improve the performance of the system by having the advantage of machine learning classifier and rule based approach together. We evaluated each component on the same dataset to see the difference on their performance. Based on our assumption the performance evaluation results in each approach increases scores of precision recall and F-measure as we move from the former machine learning classifier based to the last hybrid approach. The following table and chart describes the evaluation results of event extraction system using different approaches.



	Machine learning classifier	Rule Based Approach	Hybrid Approach	
■ **Precision**	**86.6**	**97**	**97.7**	
□ **Recall**	**79.8**	**95.5**	**96.3**	
▨ **F-measure**	**83.1**	**96.2**	**96.99**	

Figure 6-1 comparison of event extraction approaches evaluation results using chart

6.1.4. Evaluation of temporal information extraction component

Temporal information extraction is another important component of this research work. Evaluation of this component is aims to see the performance of rules to extract temporal expressions from unstructured Amharic text. We used the same data set as event extraction component to evaluate the performance of this component. Since our approach for this component is rule based including regular expressions and list of temporal keyword gazetteers. Rules are applied, using syntactic features and context of the words from their neighbors with their corresponding tags. Whereas regular expressions used to match the regular date time expressions and list of gazetteers are comprise of temporal trigger keywords. The test data set is just normal unstructured text. We used a total number of 2548 tokens, from those temporal expressions extracted as true positives are 44 and false positives are 8, false negatives are 5, and true negatives are 2491. The following contingency table refers to summarize those values obtained using the rule based approach.

Table 6-11 contingency table for temporal information extraction

	Temporal expression	**Non-temporal expression**
Temporal expression	44	8
Non-temporal expression	5	2491

From the contingency table we can deduce that the value of false negative is 5, which implies the system extracts well the actual temporal expression based on prespecified rules. Whereas the value of false positives are 8,which implies the system misses detecting some temporal expressions, but they not exaggerated even though they are actually temporal. This is due to the language natures, in which the temporal expression appears in different formats beyond our limit. The other point is the true positive and true negative values, by default the value of true negatives is always higher than the other, which indicates tokens those are out of the target groups and actually they are non- temporal expressions. True positive values indicate the actual temporal expression which is correctly extracted by the system.

To evaluate the performance of this particular approach we need to calculate the usual precision, recall and f-score using their standard formulas described in Section 2 of this document

Precision= (true positives)/ (true positives + false positives)

=44/ (8+44)*100

=0.846*100 **= <u>84.6%</u>**

Now, recall is calculated as follows:

Recall= (true positives)/ (true positives + false negative)

=44/ (44+5)*100

=0.897*100 = **<u>89.7%</u>**

Finally Fl-score is the weighted average of the precision and recall values and it and calculated as follows:

F-score= (2* p*r)/ (p+r)

= (2*84.6*89.7)/ (84.6+89.7 = **87.1%**

Table 6-12 summary of rule based temporal information extraction component

Approach	**Precision**	**Recall**	**F-measure**
Rule based temporal information extraction	84.6%	89.7%	87.1%

6.2. Result and Ddiscussion

In this work we conducted a total of six differnt experiments to evaluate the performnce of ETIEA. From those experiments the first three are for the machine learning classifier based event extraction component of ETIEA. We conducted them on a total of 10465 tokens as training and testing dataset using 10 fold cross validation. Each of the three experiments is different scenarios with a combination of features to derive the best result using different algorithms. We applied three machine learning classifier algorithms LIBSVM, Naïve Bayes, and Decision Tree on each separate scenario and we analyzed and compared the corresponding result. Among the algorithms under the specified cases the Naïve Bayes algorithm outperforms better on all feature combination scenario, with a precision, recall and F-measure values of 0.866, 0.798, and 0.831 respectively. The machine learning classifier experiments result is mainly depend on the Pos features. As the weka subseteval attribute selection method suggests the best attribute which helps the classifier to correctly predict the class of the instances is pos of the corresponding words.

The fourth experiment is conducted to evaluate the rule based event extraction component of ETIEA. Our special motivation which triggers us to come up with this approach is to resolve the deverbal entities events missed from the machine learning classifier. Our assumption is to improve the performance of the system by resolving the limitation of the former approach. We successfully achieved our assumption when we conducting the experiment on the same dataset as the former experiments we get result of precision, recall and F-measure of 97%, 95.5, and 96.2 respectively. So we see valuable difference in improvements of the accuracy, coverage and weighted average of the system from the machine learning classifier approach to the rule based approach.

The fifth and the least but not the last experiment for event extraction component is conducted to evaluate the hybrid approach event extraction component of ETIEA. Basically in this work our intension is to correctly extract events from unstructured Amharic text. Beside this objective improving the system accuracy by proposing different approach is our effortful contribution in this work. To do so, after all having the result of the pure machine learning classifier based event extraction and rule based event extraction independently. We propose another hybrid approach to get the advantage of rule based approach in conjunction with the machine learning based approach. We incorporate some heuristics on the machine learning classifier result to resolve deverbal entities missing problems. We said that the machine learning classifier is robust in detecting verb triggered events than the rule based approach. So having this some heuristics are developed and incorporated with the classifier results. The results of this experiment shows little improvement of the accuracy of the system from the previous approaches with a values of precision , recall, and F- measure of 97.7%, 96.3% and 96.99% respectively.

The six and last experiments of our work is conducted to evaluate the performance of rule based temporal information extraction component of ETIEA. This component is another standalone component to extract temporal information from unstructured Amharic text using rules, regular expressions, and context of words, gazetteer lists, and pos tagger. We don't need to apply intensive approaches like that of event extraction components because of the temporal information in a text are not as huge as other information's. To do so, we thought applying rules will be enough to resolve them. We used a little bit temporally rich dataset to conduct the experiment and we get result of precision, recall, and F-measure of 84.6%, 89.7%, and 87.1% respectively. In this study we notice that the result of the temporal information task is relatively lower accuracy in comparison with the event extraction task. It's because of that one reason could be the form of the variety of form of temporal expression appearance in Amharic text including Arabic, Geez, and Alphanumeric. Although, we tried to normalize and convert the geez numbers to Arabic numbers to avoid confusion still there is a need due to its complexity. The other issue is due to time lack of temporally rich source data we only applied the rule based methods other than using statistical approaches. To resolve this extending the sophisticated handcrafted rules by considering forms, contexts and appearance of the temporal expression in Amharic text will bring good result.

CHAPTER SEVEN

7. CONCLUSION, CONTRIBUTION AND RECOMMENDATION

In this chapter we try to address the brief summary of this study including the main contribution limitation and future works that could be extended from this work.

7.1. Conclusion

The drastic increase of huge volume of data on the web becomes a headache to search and retrieve by extracting required information. Information Extraction was created and developed to extract pre-specified information from the raw text in order to organize it in a structured format.

Event and temporal information extraction is one of the tasks of information extraction which gets attention in the recent years unlike the entity extraction which has been done for many years till now. There are some works done in Temp-Eval challenges under different languages with different approaches. In Ethiopia for languages under NLP tasks such as entity recognition and general IE model, which is experimented on entities, has been done by different researchers for the last ten years. But, still now as far as our knowledge there is no work for event and temporal information extraction in any of local languages.

The purpose of this research work is to contribute for the development of event and temporal information extraction system for Amharic. In this work we develop a generic model for event and temporal information extraction system using different approaches. Which contains major components such as preprocessing component, event extraction using machine learning classifier, event extraction using rule based approach, event extraction using hybrid approach and temporal information extraction using rule based approach. Each component is comprised of their own subcomponents and developed algorithms. We used a java programing as a developmental tool and other jar files such as publically available decision tree based part-of-speech tagger named TreeTagger, python based morphological analyzer for amahric named HornMorpho, and open source tool weka which contains the machine learning algorithm classifiers.

We conducted a total of six of experiments for each component using different scenarios. The experiments are conducted in a total of 19,486 words dataset. We analyzed the results of each experiment on each approaches and draw a comparison. Finally, in the discussion Section we clearly explained the experimental results of each model.

7.2. Contribution of the work

- Generic model is designed with different approaches for Amharic event and temporal information extraction
- Event extraction system is developed using open source weka Machine learning classifiers in java
- Rule based event extraction system from amahric text is developed
- Event extraction system for Amharic text using Hybrid approach developed
- Ethiopic numerals to Arabic number conversion
- Rule based temporal information extraction from Amharic text is developed
- Algorithms are developed for the preprocessing module, rule based approach, and hybrid approach.
- A prototype system for ETIEA is developed.
- We conducted different experiments to demonstrate the performance and accuracy of each approach for event extraction system
- The hybrid approach result shows better accuracy for event extraction component than the other two approaches

7.3. Recommendation

In this study we showed that different approaches are employed to extract events and temporal information from Amharic text. Event extraction from Amharic text using hybrid approach achieved better accuracy over rule based approach and machine learning classifiers. On the other hand the rule based temporal information yields promising result. To improve the performance of event and temporal information extraction from Amharic text one has to add extends rules by considering forms, and contexts of word for the rule based approach. For the machine learning approach increasing the feature set will help the classifier to correctly predict instances of their class. Whereas to improve the overall system performance considering semantic features of words such as wordnet, semantic role labeling, and having sentence parser helps to detect phrase level event argument extraction. To develop a full-fledged event and temporal information extraction for Amharic text further works required in the field. We recommend the following tasks as future work:

- The machine learning classifier based event extraction component we have built has low accuracy in comparison with the rule based and hybrid approach and from the literatures reviewed for machine learning approach data is crucial and adding best feature set will improve the accuracy of the system.
- Due to limitation of time and lack of readymade sentence parsers we only focus to extract events and temporal information. But, relation extraction between event and time is prominent task in this field to extend this work for the decision support system.
- We used unstructured text as training and testing set, so that extracting relation between events with the document creation time will be tough. Rather applying news style document would help to extract relation between times and document creation time
- The other important point is event argument extraction, arguments can act as nominal which participate on the target event so identifying them is another task which needs dependency parser.
- To enhance the performance of temporal information extraction adding complex regular expression and rules will yield good result.
- Employing a new scoring scheme for event ranking gives weightage based on the priority level of the events, which include the occurrence of the event in the title of the document, event frequency, and inverse document frequency of the events.

References

[1]. S. Sarawagi, Information Extraction, Foundations and Trends in Databases Vol. 1, No. 3, Indian Institute of Technology ,2007

[2]. F. Hogenboom, F. Frasincar and U. Kaymak, “An Overview of Event Extraction from Text”, Erasmus University Rotterdam.

[3]. V. Raghavendera, “DE-IIITH at SemEvalTask 12: Extraction of Temporal Information from Clinical documents using Machine Learning techniques”, Proceedings of SemEval-2016, pages 1237–1240, San Diego, California ,2016

[4]. G. Prashant, H. Sarojadevi, N. Chiplunkar, "Rule-Knowledge Based Algorithm for Event Extraction”, International Journal of Advanced Research in Computer and Communication Engineering, Nitte, Karnataka, India, Vol. 4. , 2015

[5]. F. Schilder and C. Habel**, “**From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages”, Department of Informaics, University of Hamburg, Germany.

[6]. P. Jindal and D. Roth, “Extraction of events and temporal expressions from clinical narratives”, Journal of Biomedical Informatics, United States, 2013

[7]. R. Derczynski, “Automatically Ordering Events and Times in Text”, the University of Sheffield, UK. , 2016

[8]. A. Kumar, A. Ekbal, and S. Bandyopadhyay, “A Hybrid Approach for Event Extraction”, *Polibits*, 2012

[9]. E. Peter and N. Pierre, “Using Semantic Role Labeling to Extract Events from Wikipedia”, Department of Computer science, Lund University

[10]. T. Mikiyas, Amharic Named Entity Recognition using A hybrid Approach, “MSc Thesis, Addis Ababa University, Addis Ababa” , 2014

[11]. T. Ruet, “Syntax and Parsing of Semitic Languages”, Uppsala University.

[12]. J. Strotgen and M. Gertz, “A Baseline Temporal Tagger for all Languages”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 541–547, Heidelberg University, Germany, 2015

[13]. F. Hogenboom, F. Frasincar and U. Kaymak, “A Survey of event extraction methods from text for decision support systems, ELSEVIER, Decision support system, Netherlands” , 2016

[14]. L. Tian, W. Ma, and Z. Wen, “Automatic Event Trigger Word Extraction in Chinese Event”, Journal of Software Engineering and Applications, 2012, 5, 208-212

[15]. L. Derczynskl. , J. Strotgen, D. Maynerd. and A. Mark, “GATE-Time: Extraction of Temporal Expressions and Events”, University of Sheffield, UK.

[16]. D. Jurafsky & James H. Martin, 2009, “Speech and language processing, An Introduction to Natural Language Processing”, Computational Linguistic, and Speech Recognition, 2nd Edition, Pearson Prentice Hall, New Jersey.

[17]. E. Biniyam, Y. Baye and Y. Miyayo, “Morpho syntactically annotating Amharic document to build Amharic Treebank”, Addis Ababa University, Addis Ababa, 2016

[18]. A. Beatrice, C. Vincent, T. Xavier, and V. Anne**, “**Supervised Machine Learning Techniques to Detect *TimeML* Events in French and English”, Springer International Publishing, 2015, France.

[19]. S. Andrea and G. Robert, “Annotating Events and Temporal Information in Newswire Texts”, Department of computer science, UK.

[20]. ፕሮፌሰር ባየ የማም, 2004, አጭርና ቀላል የአማርኛ ሰዋስው

[21]. L. Xiao and S. Daniel, “Temporal Information Extraction”, Association for the Advancement of Artificial Intelligence, 2010.

[22]. S. Sarawagi, “Information Extraction, Foundations and Trends in Databases”, Mumbai, Vol. 1, No. 3, 2008

[23]. P. Jakub and Y. Roman, “Multi-source, Multilingual Information Extraction and Summarization”, Theory and Applications of Natural Language Processing, Springer-Verlag Berlin Heidelberg, 2013

[24]. T. Jordi, A. Alicia and C. Neus, “Adaptive Information Extraction”, TALP Research Center, Spain.

[25]. Y. Akane, T. Yuka and M. Yusuke, “Event extraction from biomedical papers using a full parser”, Pacific Symposium on Biocomputing 6:408-419 (2001), University of Tokyo, japan

[26]. M. Atkinson, J. Piskorski, H. Tanev, E. van der Goot, R. Yangarber, V. Zavarella, Automated event extraction in the domain of border security, User Centric Media User Social Informatics and Telecommunications Engineering, Springer 2009.

[27]. M. Makoto, S. Rune and K. Jin-dong, “Event extraction with complex event classification using rich features”, J. Bioinform Boil, UK, 2010

[28] J. Alan, “Brundlefly at SemEval-2016 Task 12: Recurrent Neural Networks vs. Joint Inference for Clinical Temporal Information Extraction”, Proceedings of SemEval-2016, pages 1274–1279,San Diego, California, June 16-17, 2016.

[29] W. Bekele, “Information Extraction from Amharic language Text: Knowledge-poor Approach”, MSc Thesis, Addis Ababa University, Addis Ababa, 2014.

[30] Shubin Zhao, “Information Extraction from Multiple Syntactic Sources”, New York University, 2004

[31] M. Kaufmann, Proceedings of the Sixth Message Understanding Conference (MUC-6), 1995.
[32] Proceedings of the Seventh Message Understanding Conference (MUC-7), 1998.

[33] V. Sumithra, L. Danielle, A. Samir, C. Lee and W. Wendy, “BluLab: Temporal Information Extraction for the 2015 Clinical TempEval Challenge”, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 815–819, Denver, Colorado, June 4-5, 2015

[34] Y. Versley, “A Modular Toolkit for Co-reference Resolution”, a hand book on co-reference resolution

[35] A. Waterberg, “TempEval-2 task A: Temporal Information Extraction using Regular Expressions”, Lund Institute of Technology, Sweden, 2009

[36] K. Abdulrahman, V. Sumithra, and M. Stephane ,“UtahBMI at SemEval-2016 Task 12: Extracting Temporal Information from Clinical Text”, Proceedings of SemEval-2016, pages 1256–1262, San Diego, California, June 16-17, 2016

[37] S. Jannik and G. Michael, “HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions”, Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 321–324, Uppsala, Sweden, 15-16 July 2010

[38] M. Wondwossen and G. Michael, “Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming”, Workshop on Language Technology for Normalization of Less-Resourced Languages (SALTMIL8/AfLaT2012)

[39] D. Lars and L. Ulf, “Extracting and Aggregating Temporal Events from Text”, International World Wide Web Conference Committee (IW3C2), April 7–11, 2014, Seoul, Korea

[40] A. Mohammad and Q. Omar, “Knowledge-based Approach for Event Extraction from Arabic Tweets”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No.6, 2016

[41] A. Mohammad and Q. Omar “A Supervised Machine Learning Approach for Events Extraction out of Arabic Tweets”, semantic web journal, January 17, 2017

[42] B. Tesfaye, “Automatic Morphological Analyzer for Amharic”, MSc Thesis, Addis Ababa University, Addis Ababa, 2002

[43] E. Douglas and J. David, “Introduction to Information Extraction Technology”, Artificial Intelligence Center, SRI International

[44] G. Bjorn, “Tagging and Verifying an Amharic News Corpus”, Workshop on Language Technology for Normalization of Less-Resourced Languages (SALTMIL8/AfLaT2012), Norway

[45] L. Xiao and S. Daniel “Temporal Information Extraction”, Association for the Advancement of Artificial Intelligence, Washington, 2010

[46] H. Sinatyehu, “Information Extraction System for Amharic Text”, International Journal of Computer Science Trends and Technology (IJCST) – Volume 5 Issue 2, Mar – Apr 2017

[47] T. Getasew, “Information Extraction Model from Amharic News Texts”, MSc Thesis, Addis Ababa University, Addis Ababa, 2010

[48] M. Tesfu, “Event and Temporal information extraction”, MSc Thesis, Jimma University, 2016

[49] A. Lars, A. Atelach, G. Björn and S. Magnus, “Applying Machine Learning to Amharic Text Classification”, Stockholm University

[50] S. Roser, “temporal expressions”, 2006

[51] T. Jordi, A. Alicia and C. Neus “Adaptive Information Extraction”, TALP Research Center, Spain

[52] C. W. Isenberg, “Grammar of the Amharic Language”, London: Richard Watts, 1976.

[53] K. Judith and M. Kan “The Role of verbs in document analysis”, COLING-ACL 1998, pp. 680-686, Montreal, Canada.

[54] M. Gasser, “A Dependency Grammar for Amharic”, Indiana US5

[56] L. Yunyao and K. Rajasekar,” Regular Expression Learning for Information Extraction”, Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 21–30.

[57] D. Girma, “Manual Annotation of Amharic News Items with Part-of-Speech Tags and its Challenges”, Ethiopian Languages Research Center of Addis Ababa University, (2006).

[58] G. Kaur**,** “Usage of Regular Expressions in NLP”, International Journal of Research in Engineering and Technology, Volume: 03 Issue: 01, Jan-2014

[59] K. Schluter, “Amharic Internal Reduplication and Foot Structure: A Word-Based Approach”, Kansas Working Papers in Linguistics, Vol. 30 (2008), p. 287

[60] P. Mazur ,and R. Dale, “A Rule Based Approach to Temporal Expression Tagging”, Proceedings of the International Multi conference on Computer Science and Information Technology pp. 293–303,ISSN 1896-7094,2007

[61] J**.** Tourille, and O. Ferret, “Temporal information extraction from clinical text”, Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 739–745, Valencia, Spain, April 3-7, 2017.

[62] Haddis Alemayehu, ፍቅር እስከ መቃብር, 1958

[63] F. Schilder and C. Habel, “From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages”, University of Hamburg, 2001

[64] http://www.geez.org/#e-books

[65] A. Dillman, “Ethiopic Grammar”, 1907

[66] N. Chincher and B. Sendheim, “MUC-5 Evaluation metrics”, Science Applications International Corporation 10260 Campus Point Drive

[67] G. Micheal, “HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya”, Conference on Human Language Technology for Development, Alexandria, Egypt, 2-5 May 2011.

[68] K. Ming Leung, “Naive Bayesian Classifier”, November 28, 2007

[69] J. Teevan, “Tackling the Poor Assumptions of Naive Bayes Text Classifiers”, Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[70] K. Sudo, “Unsupervised Discovery of Extraction Patterns for Information Extraction”, PhD thesis, New York University, New York, September 2004

[71] R. Torrecampo, “Notes on Punctuation”, Philippine Content Development 2011

[72] N. Bel, M. Coll, and G. Resnik, “Automatic Detection of non-deverbal Event nouns for Quick Lexicon Production”, Proceedings of the 23rd International Conference on Computational Linguistics, pages 46–52, Beijing, August 2010

Appendices

Appendix A: Summary of related works

Ref.	Approach used		Description
	Class	Method	
[47]	Data-driven	Supervised Machine Learning Classifier (DT,SMO.NB)	Information Extraction Model from Amharic News Texts Tasks ➢ Extracting entities from Amharic infrastructure news text Limitation ➢ Feature set selection; the features used in this work are not mean to valuable for entity extraction. considering NER, POS might be crucial
[18]		Supervised Machine Learning Technique (CRF, DT,kNN)	Supervised Machine Learning Techniques to Detect *TimeML* Events in French and English Tasks ➢ extracting events from English and French TimeML annotated documents ➢ Features used includes pos, word-form and lemma ➢ Best feature set combination Limitation ➢ They solely relay on the machine learning algorithms ,if they could incorporate rules they could find good results ➢ External features used are domain dependent and.
[3]		Supervised Machine Learning	CDE-IIITH at SemEval-2016 Task 12: Extraction of Temporal Information from Clinical documents using Machine Learning techniques

		Technique (SVM,CRFDNN)	Tasks ➢ Sequence labeling ➢ Classification ➢ Relation extraction ➢ Word embedding representation which is better than previous statistical approach word representation schemes like skip-gram model which leads dense matrix multiplication.
[37]	Knowledge based	HeidelTime temporal tagger ,UIMA framework	HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions Tasks ➢ Extraction of temporal information extraction ➢ Normalization of temporal information expression
[35]		Hierarchical Regular expressions	Temporal Information Extraction using Regular Expressions Tasks ➢ Temporal expression extraction ➢ Categorization and normalization of temporal expression
[40]		unsupervised rule-based technique	Knowledge-based Approach for Event Extraction from Arabic Tweets Tasks ➢ Event extraction ➢ Named entities disambiguation of event related entities ➢ Populating extracted entities and event related entity to a knowledge base linked open data(LOD)
[28]	Hybrid	Recurrent neural network, Inference rules	Brundlefly at SemEval-2016 Task 12: Recurrent Neural Networks vs. Joint Inference for Clinical Temporal Information Extraction Tasks ➢ Sequence labeling

			Identifying span of event and time expression ➢ Classification Relation between event and document creation time
[33]		ClearTK, SVM, CRF and Rule based Approach	BluLab: Temporal Information Extraction for the 2015 Clinical TempEval Challenge Tasks ➢ event span detection ➢ TIMEX3 and event attribute classification ➢ document relation time and ➢ narrative container relation classification Limitation ➢ For their narrative container relation tasks their searching mechanism to get the possible pair relations are exhaustive search which increases false positives and it affects the overall system performance.
[25]		Full parser, General purpose grammar approach and domain specific mapping rules	Event extraction from biomedical papers using a full parser Tasks ➢ Event and its argument extraction Limitation ➢ Shallow parser is effective than full parser to reduce the lexical ambiguity as well as to easily detect relation and reaction of events in a complex sentence.

Appendix B: Non-Deverbal Entities

አደጋ

ሰርግ

ሃዘን

ደስታ

ሽኝኝት

ጉብኝት

ስህተት

ትግል

ውሳኔ

ልደት

ጥቃት

ስለላ

ትኩረት

ፈጠራ

ውጤት

ሩጫ

ውድድር

ስምምነት

ውል

ሞት

ስብሰባ

መጓጓዝ

መግደል

ማጥፋት

ምርጫ

ደሞዝ

ሰራተኛ

ጉብኝት

ጦርነት

ሙከራ

ፍተሻ

ግድያ

ምኞት

ስብሰባ

ማመስገን

ማንበብ

መንገድ

መረጃ

ግንዛቤ

ፍልስፍና

ችግር

ቁጥጥር

እውቀት

ችሎታ

ፍቅር

ፍጥረት

እንዲያውም

ዉሳኔ

ድርጊት

ገበያ

ተማሪ

እንጨት

ዉድድር

ስርጭት

መዝናኛ

መግቢያ

ዕድል

አፈጻጸም

መብረር

ግንኙነት

መልእክት

ሞት

ዉይይት

ምክር

ደም

ሙከራ

ጠቃሚነት

ግምገማ

ክፍያ

ትግበራ

ከተማ

ጥልቀት

ስብስብ

ጭንቀት

ምንጭ

ቅንብር

ግንኙነት

መግለጫ

ትዕግሥት

ማስተዳደር

መልስ

ምርጫ

ትርጉም

ስምምነት

ኪሳራ

ሥራ

ችግር

መከራ

ማነጻጸር

መደናገር

ግንባታ

መሪነት

ፈቃድ

ማቅረቢያ

ማስተዋወቅ

ዘፉን

Appendix C: Amharic Stop words

እንዲሁ
ሁሉም
ሁኔታ
ሆነ
ሆኑ
ሆኖም
ሁል
ሁሉንም
ላይ
ሌላ
ሌሎች
ልዩ
መሆኑ
ማለት
ማለቱ
መካከል
የሚገኙ
የሚገኝ
ማድረግ
ማን
ማንም
ሰሞኑን
ሲሆን
ሲል
ሲሉ

ስለ
ቢሆን
ብለዋል
ብቻ
ብዛት
ብዙ
ቦታ
በርካታ
በሰሞኑ
በታች
በኋላ
በኩል
በውስጥ
በጣም
ብቻ
በተለይ
የተለያየ
የተለያዩ
ተባለ
ተገለጸ
ችግር
ታች
ትናንት

ነበረች
ነበሩ
ነበረ
ነው
ነይ
ነገር
ነገሮች
ናት
ናቸው
አሁን
አለ
እስካሁን
እስከ
እባክህ
እባክሽ
እባክዎ
አንድ
አንጻር
እንኳ
እስከ
እዚሁ
እና
እንደ
እንደገና

ወቅት
እንጂ
እዚህ
እዚያ
ከኋላ
ከላይ
ከሰሞኑ
ከታች
ከውስጥ
ከጋራ
ከፊት
ወዘተ
ወይም
ወደ
ዋና
ወደፊት
ውስጥ
ውጪ
ያለ
ያሉ
ይገባል
የኋላ
የሰሞኑ

የታች
የውስጥ
የጋራ
ይህ
ደግሞ
ድረስ
ጋራ
ግን
ጥቂት
ፊት
ደግሞ
ጋር
ሲሉ
ብለዋል
ስለሆነ
አቶ
ሆኖም
ይናገራሉ
ተጨማሪ

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: Ephrem Tadesse

Signature:

Date: October 20, 2017

Confirmed by advisor:

Name: Mr. Debela Tesfaye (PhD Candidate)

Signature:

Name: Mr. Tesfu Mekonen

Signature

Date: October 20, 2017