ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Event Modeling from Amharic News Articles

Bekele Abera Hordofa

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

Addis Ababa, Ethiopia

February 2018

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Event Modeling from Amharic News Articles

Bekele Abera Hordofa

Advisor: Fekade Getahun (PhD)

This is to certify that the thesis prepared by Bekele Abera, titled: *Event Modeling from Amharic News Article* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Fekade Getahun (PhD)	_____	_____
Examiner:	Solomon Gizaw (PhD)	_____	_____
Examiner:	Yaregal Assabie (PhD)	_____	_____

Abstract

With the increase of online information overload available to us, everyday tasks such as extracting, searching, understanding and relating relevant data have become intractable. A large percentage of these information are discussing past, current, and future real world events. Events are dynamic data structures that play a key role in understanding phenomena happing in real world, which are basically driven by the four 'W's' (what, who, when, and where). This natural progression of questions is a classic example of what one might ask about an event. This results in natural way to explain complicated relations between people, places, actions and objects. Event centered modeling captures the dynamic aspects of an event along with semantic representation of event facts.

In this research work, we have proposed and developed event extraction and representation model from Amharic news article. Event modeling involves key event identification, event elements extraction, and event semantic elements representation. Event triggers tells the action taking place in news article. To identify the mention of event in news article we used manually collected event trigger words and phrases from various news domains. For event elements extraction, we used named entity recognizer and other local features like potential trigger, event extent, path from the extent to head word of the trigger. Machine learning Maximum entropy classifier is trained using event related news article collected from Fana Broadcast Corporate news archive. For event representation, we designed ontology based event representation model that provides deeper semantic through event information representation.

A prototype showing an event extraction and representation is developed using different programming environment. Evaluation of trigger identification and event elements extraction is carried out by comparing manually tagged news article with the automatically extracted event information by the system. The evaluation result shows that the trigger identifier module obtain precision (67.1%) of event correctly which contributes to the better event elements extraction. The event elements extractor component shows greater obtaining precision (69.1%) while event classification module classify about (72%) of event correctly. The representative ability of our event representation model is evaluated with respect to requirements and event dimensions we covered in this work.

Keywords: Event, Event Modeling, Event Trigger, Event Elements, Event Representation

Dedication

To my Mother Fitale Gonfa (ፊታሌ ጎንፋ)

For your unconditional love and support. May God give you a long and healthy life.

Acknowledgments

First and foremost, I would like to thank the almighty God (ጌታ) for giving me strength and determination to finish this thesis work.

Second, I would like to express my sincere gratitude to my advisor, Dr. Fekade Getahun, for his patience, motivation, immense knowledge and continuous support during this thesis work.

Thanks to my family (my mom, my dad, my brothers and my sister) for your endless support for all these years. You were always supporting me in every way.

Lastly my thanks goes to my fellow friends, teachers and others who has contributed to the successful accomplishment of this thesis work.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms and Abbreviations

EE	Event Extraction
EM	Event Modeling
EO	Event Ontology
IE	Information Extraction
IR	Information Retrieval
KIF	Knowledge Interchange Format
LODE	Linking Open Descriptions of Events
NED	New Event Detection
NER	Named Entity Recognition
NLP	Natural Language Processing
NOEM	New Ontology Event Model
OWL	Web Ontology Language
POST	Part of Speech Tagger
RDF	Resource Distribution Framework
RED	Retrospective news Event Detection
SEM	Simple Event Model
SHOE	Simple HTML Ontology Extensions
SW	Semantic Web
SWOOP	Semantic Web Ontology Overview and Perusal
TDT	Topic Detection and Tracking
XML	eXtensible Markup Language
YAGO	Yet Another Greater Ontology

Chapter One

Introduction

1.1 Background

Nowadays, thousands of Amharic news articles written and published every day over the web. A large percentage of these articles are discussing current, past and future real world events. The notion of event has been widely used in many research fields like natural language processing, information retrieval and information extraction [1]. There is no generally accepted definition of an event, it is something unusual, fresh, something that people have not heard before and crucially, is of interest to readers like car accident, meeting, natural disaster, *etc.* According to Miller *et al.* [2], a general definition of event is something that happens or occur at a given place and time. In topic detection and trucking, an event is defined as something that happens at a given place and time, along with all the necessary preconditions and unavoidable consequences [3].

Event extraction is high level information extraction task which tries to formulate an event as who did what to whom, when and where [1, 4]. Most news and microblogs in the world give priority to events in their country, unless a very important event happens in another part of the world, because people want to learn what is happening around them. Events are described using trigger words and event elements. The task of event extraction is to automatically identify events in free media and to derive detailed information about them, ideally identifying who did what to whom, when and where.

Event elements define what happened during the seminal event, who was involved (people, organization) during the course of event, when and where the seminal event happened. The rule of 4W originally states that event story should be considered as complete if it answers the checklist of 4W [1]. The 4W are considered to be rich enough for user to understand the whole event story. Event extraction is responsible for finding the most informative terms which describe the event that occurred, and it is crucial way in event management, intelligence gathering, and decision making [6, 7]. Various techniques are proposed for event extraction, according to the type of event, detection task, and detection methods. According to the detection task and target application, classified into retrospective event detection and new event detection techniques. Depending on the available information on the event of interest, event detection can be classified into specified (planned) and unspecified (planned) techniques [5].

Manually extracting event information from a large amount of unstructured document is a very tedious and time consuming task [8, 9]. For this purpose, event extraction models have been proposed to help the extraction of event information from news articles [1, 9]. Extracting and representing event information is essential way to build event knowledge base [10].

1.2 Motivation

With the increase of online information overload available to us, everyday tasks such as extracting, searching, understanding and relating relevant data have become intractable. Event extraction and representation from a variety of media (news, social media) is highly demanding in government, news publishers, and other service provider organizations. One of the recognized causes for this issue is the semantic gap existing between our conceptualizations of the world, usually expressed using language or other high level abstractions, and our experience of the world. People observe and understand the world through event because it is a suitable unit in accordance with aspect of human perception.

In order to extract event facts from a specific event scenario described in news article, the textual events belonging to the event scenario are mapped to the events of the model describing the news article. Most general purpose event extraction is designed or developed keeping English language characteristics in mind. It is logical to ask question such as:

- Does general purpose event extraction model developed for other languages like English work for Amharic language?
- Does existing event representation or event representation model developed for English and other languages work for Amharic langauge?

To answer these questions further investigation or research is highly required. Therefore due to information explosion over internet there is a need for a model that can extract event facts and semantically represent the extracted facts from unstructured Amharic news articles on the web. Several practical applications that can be arise from a good solution to event modeling. Currently, this task is performed manually by media analysts, and online digital news editor, who have the task of collecting, interpreting, and presenting news from multiple news sources. In addition, a system that could organize events automatically would be usefull for news applications where decision making process is based on new events and the evolution of existing events.

1.3 Statement of the Problem

In todays real world, a number of planned and unplanned events can happen or occur every day. These events are presented or published in the form of news over news broadcast services or on social media networks, that is unstructured and heterogeneous events over dynamic web. At the same time, the number of internet users, web news articles that are writen in Amharic language online news broadcast, micblog service providers, *etc.*, are leading to information explosion over the web. News service providers are pushing articles to the users. As a result, users have to read the whole content of news article to understand the concept of event story. Thus tools that can extract event facts to satisfy users demand are becoming critical.

Automatically identifying news article associated with seminal events is a challenging problem due to the heterogeneous and unstructured nature of news article. Seminal event is something new wich is interesting to public. In existing system seminal events are described in free text or unstructured text like mundane events or any other ordinary news articles. Seminal event are more accessible and understandable, if event information are extracted and semantically represented, stored in event knowledge base.

Event scenario is described using event elements. Event elements provide multi-dimensional semantic understanding of event description. Event extraction is langauge depedent, that is event extraction system developed for English or any other language cannot work for Amharic language of the same domain. A number of event extarction model has been developed in a variety of language such as English[6], Chinise [1, 11], Turkish[3] and others. To our knowledge there is no event extraction model developed for Amharic langauge. Thus, in this work we propose event extraction model for Amharic language.

1.4 Objectives

General Objective

The main objective of this thesis work is to design event extraction and representation model from Amharic news articles.

Specific Objectives

In order to achieve the above general objective, the following specific objectives are identified:

- Review related literature in the area of event modeling,
- Study specific characteristics of Amharic language,
- Adopt or select best tools, techniques and approaches for event modeling,
- Identify components of event modeling,
- Prepare Amharic event triggering lexicon and training dataset,
- Design event extraction and representation model,
- Develop a prototype that shows the viability of the proposed system,
- Evaluate the performance of the proposed system.

1.5 Methodology

In order to achieve the objectives of the thesis, the following methods and procedures will be used.

Literature Review

Detail literature review and assessment will be conducted on works which are related to event modeling (key event identification, event semantic elements extraction and event knowledge representation and other related areas), from the review of this work, best approaches will be used for overall thesis work.

Data Collection and Preparation

Event modeling requires large data source for event extraction process. Data sources will be collected from different sources for the development of the system. Different NLP tools will be used to process row data collected from different domains.

Design and Implementation

In the design phase of the proposed work, the overall system architecture and best algorithms will be designed or adopted. Different tools will be used to construct ontology and selected a programming language will be used to develop the prototype.

Evaluation the performance

Proper evaluation method will be used and the proposed solution will be evaluated using standard information extraction measures and selected data sets.

1.6 Scope and Limitations

Event extraction is a very complex and rigorous task which needs the understanding of the natural language and the specific domain in need. In this the research work primarily focus on event extraction and representation model from Amharic news article. The scope of this work includes extracting seminal real world events mentioned in Amharic news article through key event identification, event semantic elements (4W) extraction and event representation.

This study does not consider deep semantic relationship among events and sub-event extraction, only key event and its elements is considered. In addition to this news article containing multilingual (machine translation), image news article (OCR) and other media (video and audio) is not the concern of this study.

1.7 Application of Results

Event modeling has wide area of application in real world environment. Up on successful implementation of this research work several practical applications can arise from good solution to event extraction and representation model. The following application areas are among the basic one.

- Event monitoring in the news domain: Application that monitors the news sphere to detect interesting and emerging events, then store all news article telling about events in event knowledge base.
- Event prediction and decision making: predicting a specific events like disruptive or violent events and early decision making. Gathering information about violent events is an important task for better understanding conflicts and for developing environmental monitoring systems for automatic detection of precursors for threats in the fields of conflict and health.
- Integrating and querying event knowledge base into application like information retrieval, question answering, *etc.* can improve problem of understanding user query, because usually users uses a search query using event elements.

1.8 Organization of the Research Work

The rest of this research work is organized as follows. In Chapter Two, we present review of Amharic language and natural language processing, and discuss approaches used for event

extraction and knowledge representation. Chapter Three introduces related works to event modeling from news article and related areas. Chapter Four focuses on the design of event extraction and representation model. In chapter Five we provide implementation and evaluation results of the proposed wo3e4rk. Finally, we present conclusion and future work in Chapter Six.

Chapter Two

Literature Review

2.1 Introduction

This chapter presents a review of literatures to provide brief understanding of what has been done so far and what to be done in this thesis work. Basically the review presented in this chapter deals on characteristics of Amharic language, natural language processing, information extraction, ontology modeling technologies, event modeling approaches, task of event extraction and knowledge representation are discussed.

2.2 Amharic Language

2.2.1 Amharic Writing System

Amharic is one of the Semitic language spoken in many parts of Ethiopia and it is an official working language for Federal Democratic Republic of Ethiopia [13]. It is also a working language of several regional states like Amhara, South Nation and Nationalities and *etc.* thus it have nationwide coverage. Amharic is written using a writing system called fidel, adapted from the one used by Ge'ez language. This unsystematic borrowing from Geez has resulted in redundant characters in the Amharic fidel. These create different symbol which have the same pronunciation. Although these different fidels give each word different meaning in Geez, while in Amharic language they have been used interchangeably. These fidels ሀ, ሐ, ኀ and ኻ all pronounced as (hä), ሰ and ሠ refers (sä), አ and ዐ refers (a) and, ጸ and ፀ refers (tsä). Amharic took the whole Geez alphabet (fidel) and uses it in the Amharic writing system [14]. Amharic language writing system contains 34 base characters each of which occurs in a basic form and six other forms known as orders [13]. The seven orders represent syllable combinations consisting of a consonant following vowel. The vowels are fused to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change their order. In Amharic writing system there is no capital-lower case distinction. The 34 basic characters and their orders give 238 distinct symbols. In addition to the 238 symbols, there are other symbols with a special feature usually representing labialization, for example ኳ, ሟ, ቷ and *etc*.

Amharic has its own way of writing having numerals and punctuations [14]. In Amharic language, different punctuation marks are used for different purposes. Most text processing works with

sentence and word based unit therefore larger blocks of text such as paragraphs or whole article is split into single sentences and sentences. It starts with a sequence of characters to identify the elementary parts of natural language such as words, punctuation marks and separators. In the old writing system, a colon (፡) has been used to separate two words. Nowadays the two dots are replaced with whitespace. An end of a statement is marked with four dots (።) while netal serez (፣) is used to separate lists or ideas just like the comma in English.

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. Amharic numeration system contains one-to-ten, twenty-to-ninety, hundred and a thousand symbols [13]. The remaining numerals are derived from these basic symbols. Amharic character, numerals and punctuation marks are presented in *Appendix I.*

2.2.2 Amharic Morphology

Amharic is one of the most morphologically complex language. It exhibits a root-pattern morphological phenomenon [15]. Root is a set of consonants which has a basic lexical meaning. According to Baye Yimam [16], Amharic words are categorized under five categories based on the use of morphology and position of the word in sentence. Amharic word categories are noun, verb, adjective, adverb and preposition. Nouns are words used to name or identify any class of things, people, places or ideas or a particular of these. Verbs are the most important part of speech because it shows the action or state, word that tells the listener or reader what is happening in the sentence and have more to do with mental processes and perceptions. Adjective is a word that comes before a noun and add some kind of qualification to the noun. Adverb is a word that qualifies the verb by adding extra idea from time, place and situations point of view. Preposition is a word which can be placed before a noun and perform adverbial operations related to place, time, cause and *etc*.

Amharic uses different affixes to create derivational and inflectional morpheme [15]. Derivation is achieved by affixation (prefix, infix, and suffix) or compounding. Inflection is achieved by either changing vowels or repeating consonants; and then adding the necessary affixes or suffixes. Amharic nouns can be derived from adjectives (noun ደግነት is derived from adjective ደግ), verbal roots (noun ነገር is derived by infixing vowels between consonants to verbal root ነ-ግ-ር), stems,

stem-like verbs (noun ዝምታ is derived from stem like verb ዝም) and nouns themselves (noun ልጅነት is derived from noun ልጅ). On other hand, Amharic nouns are inflected for number, definiteness, cases (objective, possessive) and gender. For example plural form of noun ተማርዎች is obtained by adding suffix -ዎች to singular noun ተማር.

Amharic verb derivations and inflections are even more complex than those of nouns and adjectives [15]. As a result of this, thousands of verbs (in surface forms) are generated from a single verbal root. Several verbs in surface forms are derived from a single verbal roots, verbal stems and compound words. For example verb ሰበር is derived from verbal root ስ-ብ-ር, verb ተሰበረ is derived from verbal stem ተበረ- and *etc.* Amharic verbs are also marked for any combinations of person, gender, number, case and tense/aspect. For example, from the verbal root ውሰድ (to take), we can derive verbal stems such as ወሰደ (he took), ወሰደች (she broke), ወሰድኩ (I broke), አልወሰድኩም (I didn't take), አልወሰደችም (she didn't take), አልወሰደም (he didn't take), አልወሰደኝም (he didn't take me), *etc.*

Amharic adjectives can be derived from verbal roots, nouns, stems, and compound words [15]. For example adjective word ጥቁር is derived from verbal root by infixing vowels between consonants ጥ-ቅ-ር, adjective ተራራማ is derived from noun ተራራ by suffixing bound morpheme, ደካማ is derived from stem ደካም by suffixing bound morpheme, *etc.* In addition to this Amharic adjectives are marked for any combination of number, definiteness, gender and case [15]. For example ጠባቦች is derived by adding –ኦች to singular adjective ጠባብ. Few primary adjectives (which are not derived) exist in the language [15].

2.2.3 Amharic Phrases

Phrases are syntactic structures that consist of one or more words but lack the subject-predicate organization of a clause [17]. Phrases are composed of either only head word or other words or phrases with the head combination. The other words or phrases that are combined with the head in phrase construction can be specifiers, modifiers and complements. According to Baye Yimam [16, 17], Amharic phrases are categorized into five categories, namely noun phrase (NP), verb phrase (VP), adjectival phrase (AdjP), adverbial phrase (AdvP) and prepositional phrase (PP).

Amharic noun phrases (NP) is a phrase that has a noun as its head [16, 17]. Noun phrase construction, the head of the phrase is always found at the end of the phrase. Noun phrase can be

made from a single noun or combination of noun with either other word classes including noun word class. The simplest NP consists of a single noun or pronoun such as (him), (her), (them), *etc*. A complex NP can consists of a noun head and other constituents (like complements, specifiers, adverbial and adjectival modifiers) that modify the head from different aspects [16]. The grammar rule for NP can be formulated as NP →Spec AdvP Adj NP and can further be rewritten as NP →NP N. There are many combinations for the Amharic NP constituents. Examples of noun phrase are: ቀለበት (ring), የአልማዝ ቀለቀበት (diamond ring), ትልቅ የአልማዝ ቀለበት (big diamond ring), *etc*.

Amharic verb phrases (VP) is constructed with a verb as a head, which is found at the end of the phrase, and other constituents such as complements, modifiers and specifiers [16, 17]. But not all the verbs take the same category of complement. Example of verb phrase, ሄደ (went), ትምህርት ቤት ሄደ (went to school), በመኪና ወደ ትምህርት ቤት ሄደ (he went to school by car), etc. both by car (በመኪና) and to school (ወደ ትምህርት ቤት) are prepositional phrases (PPs) modifying went (ሄደ) from manner and place points of view, respectively. Therefore, structure rule for this example VP is VP→PP PP V.

The construction of Amharic adjectival phrases (AdjP) is similar to that of a NP and a VP. Amharic Adjectival phrase (AdjP) is constructed with an adjective as a head word and other constituents such as complements, modifiers and specifiers [16, 17]. The head word is placed at the end. The structural rule governing this phrase is: AdjP →Spec Adv Adj. Examples of adjectival phrases are: ንበዝ (clever), በጣም ንበዝ (very clever), እንደ ወንድሙ ንበዝ (very clever like his brother), *etc.*

Amharic prepositional phrase (PP) is made up of a preposition head and other constituents such as nouns, noun phrases, prepositional phrases, *etc* [16, 17]. Unlike other phrase constructions, prepositions cannot be taken as a phrase, instead they should be combined with other constituents and the constituents may come either previous to or subsequent to the preposition. If the complements are nouns or NPs, the position of prepositions is in front of the complements whereas if the complements are PPs, the position will shift to the end of the phrase. Prepositional phrases is formed by the structural rule: PP →PP PP and each of the PPs on the right hand side can further be analyzed as: PP →PN. Examples of prepositional phrases are: እንደ ልጅ (like a child), ከወንዙ አጠገብ (close to the river), *etc.*

Amharic adverbial phrases (AdvP) are made up of one adverb as head word and one or more other lexical categories including adverbs themselves as modifiers [16, 17]. The head of the AdvP is placed at the end. Unlike other phrases, AdvPs do not take complements. Most of the time, the modifiers of AdvPs are PPs that come always before adverbs. Adverbial phrases are formed by the rule: AdvP→Adv or AdvP→Adv Adv, *etc.* Examples of adverbial phrases are: ክፉኛ (severely), ክፉኛ ተጋጨ (crashed severely), በጠም ክፉኛ (very severely), *etc.*

2.2.4 Amharic Sentence Construction

A sentence is a group of words that conform with the grammatical arrangement of the language and capable of conveying meaningful message to the reader. A sentence in Amharic can be a statement which is used to declare, explain, or discuss an issue [14, 16]. The combination of phrases create another phrase that can express a full idea on something is a sentence. The word order in Amharic clauses is generally subject object verb (SOV) arrangement. When Amharic sentence is viewed from grammatical structure point of view it is a combination of noun phrase and verb phrase. The noun phrase comes first and then the verb phrase. Based on the number of phrases they contain sentences in Amharic are categorized under two basic categories simple sentence and complex sentence [16]. Simple sentence only contain a single verb while complex sentence is constructed by combining more than one noun phrases and verb phrases. A simple Amharic sentence consists of an NP, which is the subject, followed by a verb phrase that comprises the predicate. Complex sentences in Amharic are those sentences that are composed of complex phrases such as NP, VP, or AdjP. The pattern of combination could take the form of a complex NP and a simple VP, a simple NP and a complex VP, or both complex NP and VP.

Baye Yimam [16], classifies Amharic sentences into four: declarative sentences, interrogative sentences, negative sentences and imperative sentences. Declarative sentences are used to convey ideas and feelings that the speaker has about things, happenings, events, feelings, *etc*. Its main objective is description of some issue. The news articles use the declarative sentence for expressing different information on different issues. Interrogative sentence is used to ask a question. A sentence that questions about the subject, the complement, or the action the verb specifies. In order to construct interrogative sentences, Amharic sentences usually involve such interrogative pronouns as ‘who’, ‘what’, ‘where’, ‘how many’, and ‘when’. These

interrogatives can then be combined with prepositions to produce some more interrogative prepositional phrases like ‘from whom’, ‘why’, etc.

2.3 Natural Language Processing and NLP Tools

Natural language processing (NLP) can be defined as the automatic or semi-automatic processing of human language. NLP deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages [19]. Understanding language means, among other things, knowing what concepts a word or phrase stands for and knowing how to link those concepts together in a meaningful way. One of the challenges inherent in natural language processing is teaching computers to understand the way humans learn and use language. NLP tools play a significant role in various research areas. The task of event identification and event semantic elements extraction requires various NLP tools like POST, NER, morphological analyzer *etc.*

Part of speech tagging is one of the fundamental processing step for any language in NLP and language automation, *i.e.*, the capability of a computer to automatically tag part of speech from a given sentence [18]. The basic processing step in tagging consists of assigning POS tags to every token in the text with a corresponding POS tag like noun, verb, adjective, adverb, preposition, *etc*., based on its definition, as well as its context. The input to a tagging algorithm is a string of words and a tag set. The output is a single best tag for each word. Words are the fundamental building block in every area of language processing; it requires extensive knowledge about words that are heavily based on the lexical knowledge. In contrast to other data processing systems, language processing applications use knowledge of the language.

Part-of-speech tagging is often considered as the first phase of a more complex natural language processing application [18]. Knowing the lexical category of a word provides a way to define events and objects that are found in a given text. Most common POS tags are [18]: Noun (N), Verb (V), Adjective (AJ), Auxiliary verbs (AUX), Numerals (NU), Adverb (AV), Punctuation (PU), *etc*. Among these verbs, nouns and adjectives that describe events as they take place are considered typical features in event detection.

Named Entity Recognition (NER) is a process of identifying and categorizing all named entities

in a document into predefined classes like person, organization, location, time, and numeral expressions [14, 21]. NER is developed as a subtask of information extraction, because people realized that information units like names including person, location and organization names, and numeric expressions including time, date, money and percent expressions are the key points for information extraction. Extraction of named entities from text is simple for humans. People firstly use orthographic rules in order to find named entities by looking at the first letter of a word. In English, if a word starts with a capital letter, then it is a candidate for a named entity.

For named entity recognition, there are two approaches from the point of view of computer: rule based approach and machine learning approach [21]. In rule-based approach, the entities are analyzed by experienced linguistics and handcrafted rules are created. In order to extract entities mainly three phases are used: Linguistic preprocessing, named entity identification and named entity classification. Machine learning approach is performed mainly in two stages: feature extraction and feature selection. In the feature extraction stage, previously generated training corpus is used. Named entity recognition is a crucial to event modeling, because the results would direct impact to the final extracted 4-tuples: person, date, location and event related keywords [22, 23]. The recognized named entities from news articles are then used to build event repository.

Morphological analyzer is an important analysis step in a number of areas such as natural language processing, information retrieval, topic modeling and text classification [15]. Morphological analyzer is commonly used as a preprocessing method to reduce morphological variants to a single feature. It is language dependent and should be tailored for each language, since languages have a varying degree of differences in their morphological properties [20]. Morphological analysis mainly aims at stripping off prefixes and suffixes, while leaving the root forms unchanged. For morphology poor languages such as English morphological analyzer is not a big issue, while for most other natural languages altering the roots (infixes), splitting compounds, handling derivate processes, *etc.*, are needed in order to perform a complete morphological analysis.

For morphologically complex languages like Amharic, the process involves dealing with prefixes, infixes and derivatives in addition to the suffixes. Morphological analyzer is widely used in information extraction, with the assumption that morphological variants represent similar meaning. To properly classify the concepts, it is also necessary to make them grammatically uniform, it is undesirable to differ between different forms of the same word.

2.4 Information Extraction

Information extraction is concerned with extraction of relevant information from text and stores them in a database for easy use and management of the data [19]. Information extraction has three major components regardless of the language and domain on which it is developed for, which are linguistic preprocessing, the learning and application stage. IE addresses the problem of information overload by locating the target phrases from document and transforms them in to semi-structured or structured representation. Automatic information extraction system is classified into rule-based, machine learning and hybrid [19, 22].

Rule-based methods are the earliest ones used in information extraction that use database of predefined and hand-crafted rules that specify knowledge typically in form of regular expressions [22]. It performs better for small or restricted application domains and short development time for a set of generally applicable and observable rules. But the problem with this approach is that manual creation of rules is expensive, and almost impossible to enlist all the rules that are required for identification and classification purposes. Since rule writing requires enormous human effort, therefore an easier approach which looks for patterns and relationships in the text to make a model using statistical models and machine learning algorithms. A great advantage of machine learning approach is its ability to be portable and maintainable with ease as compared with the rule-based approach but still it requires a huge corpus in order to build a high performing extraction system [23].

Most of the information extraction systems use the supervised learning approach or guided by external resource like ontology [24, 25]. The statistical approach uses annotated training corpus which is divided into two parts i.e. the training corpus and test corpus. The training corpus is used to train the model about the different annotation in the text and the test corpus is used to test the extraction model how much efficient it becomes after training [26]. These IE approach uses different features such as token string, capitalization, and token type (word, number, etc.). In addition, others use linguistic features such as part-of-speech, semantic information from gazetteer lists, and the outputs of other information systems (most frequently general purpose named entity recognizers). A few systems also exploit genre-specific information such as document structure. In

general, the more features the system used, the better performance it could achieve.

2.5 Ontology Modeling Technologies

2.5.1 Ontology

Ontology is a special kind of information representation, in which relevant concepts and relations of entity are considered. Wang *et al.* [12, 27], define ontology as a shared, formal and explicit understanding of some domain, which is often conceived as a set of entities, relations, functions, axioms and instances. Thus, ontologies are a means to explicitly specify conceptual models with logic based semantics. Nowadays ontology attracts many attentions in computer science, largely because of envision of semantic web, where real semantic of the web lies on ontology [28]. A. Scherp *et al.* [29] find different type and classify ontology into three layered architecture of ontology libraries such as foundational ontologies, core ontologies, and domain ontologies.

Foundational ontology is generic ontology, upper level ontology. It considers large scope and is highly reusable in different modeling scenarios. Examples of foundational ontologies are the ABC ontology [30], CIDOC-CRM [31], and *etc.* Core ontologies can be based on foundational ontologies and provide a refinement to foundational ontologies by adding detailed concepts and relations in their specific field. Examples of core ontologies are: Event Ontology (EO), Linking Open Description of Events (LODE), F-Model, Simple Event Modeling (SEM), and *etc.* Domain ontologies find representation of knowledge that is specific for a particular domain. Domain-specific ontologies can be used as external sources of background knowledge in combination with core ontologies. Examples of domain ontologies are: the foundational model of anatomy as a domain specific medical ontology describing the anatomy of the human body, the Gene Ontology[1], Music Ontology[2] , Soccer Ontology[3], and *etc*.

2.5.2 Ontology Development Tools

The development of ontology demands the use of various tools. Ontology development tools help to acquire, organize, and visualize the domain knowledge before and during the building of a

[1] http://www.geneontology.org/

[2] http://www.musicontology.com/

[3] http://dbpedia.org/ontology/SoccerPlayer

formal ontology [32]. Classes are the focus of most ontology and describe concepts in the domain. Slots describe properties of classes and instances. Developing ontology may include: selection of domain and scope, consider reuse, find out important terms, defining classes and class hierarchy, defining properties of classes and constraints, and create instances of classes.

A range of open source and commercial tools are available which assist for the development of various ontologies. These tools can be applied to several stages of the ontology life cycle including the creation, implementation, and maintenance of ontologies.

Protégé

Protégé[4] is an ontology and knowledge base editor produced by Stanford University [28]. It enables the construction of domain ontologies, customized data entry forms to enter data through graphical user interface. Protégé allows the definition of classes, class hierarchies, variables, variable value restrictions, and the relationships between classes and the properties of these relationships. Protégé is used by different users, because it provides better flexibility for meta-modeling, enables the construction of domain ontologies; customize data entry forms to enter data.

According to [28], Protégé is most widely used and domain independent ontology development tool. The strength of Protégé is that, it supports the same tool builders, knowledge engineers and domain specialists [32]. This is the main difference with existing tools, which are typically targeted at the knowledge engineer and lack flexibility for meta-modeling. This latter feature makes it easier to adapt Protégé to new requirements and/or changes in the model structure.

Semantic Web Ontology Overview and Perusal (Swoop)

SWOOP is a Web based OWL ontology editor and browser that contains OWL validation and offers various OWL presentation syntax views [32]. It has reasoning support and provides a multiple ontology environment that can be compared, edited and merged. Different ontologies can be compared against their description logic based definitions, associated properties and instances.

SWOOP's interface has hyperlinked capabilities so that navigation can be simple and easy. SWOOP does not follow a methodology for ontology construction. This is possible either by purely

[4] http://protege.standord.edu/

linking to the external entity, or importing the entire external ontology. It is not possible to do partial imports of OWL. There are several ways to achieve this, such as a brute-force syntactic scheme to copy/paste relevant parts (axioms) of the external ontology, or a more elegant solution that involves partitioning the external ontology while preserving its semantics and then reusing (importing) only the specific partition as desired.

Apollo

Apollo[5] is open source ontology editor developed by KIM (Open University) [32]. It supports standalone semantic web architecture with extendable plug-ins. Apollo's ontology storage is in file format and it do not support backup management. It support import from Apollo Meta languages and export to OCML and CLOS, but it is not interoperable with other ontology tools. Apollo does not support inference services, both built in and other attached inference engines. Usability is less compared to other ontology development tools, it don't have GUI, zooms, collaborative working with others, but it has ontology libraries.

The modeling is based around the basic primitives, such as classes, instances, functions, relations *etc.* Internal model is built as a frame system according to the internal model of the OKBC protocol. The knowledge base consists of ontology that is hierarchically organized. Ontology can inherit other ontologies and then use classes of inherited ontology's as its own. Every ontology inherits at least one ontology, a default ontology, which contains all primitive classes: Boolean, integer, float, string, list *etc.* Class contains slots of two types: non template and template slots.

WebOnto

WebOnto[6], is a tool providing web-based visualization, browsing and editing support for developing and maintaining ontologies and knowledge models specified in OCML [33]. WebODE is a scalable and integrated workbench for ontology engineering based on the ontology development methodology. It supports building ontology at the knowledge level, and translates it into different ontology languages. WebODE is based on a client-server architecture which provides high extensibility and usability by allowing the addition of new services and the use of existing

[5] http://apollo.open.ac.uk/index.html

[6] http://webonto.open.ac.uk/

services, while Protégé-2000 and OntoEdit are based on plug-in architecture, ontologies are stored in an SQL database to attain high performance in the case of a large ontology.

It has export and import services from and into XML, and its translation services into and from various ontology specification languages such as RDF(S), OIL, DAML+OIL, X-CARIN, Jess and F-Logic. In the ontology development phase, WebOnto has ontology editing service, WAB: WebODE axiom builder service, inference engine service, interoperability service and ontology documentation service. The ontology editor provides users with form based and graphical user interfaces, WAB provides an easy graphical interface for defining axioms. It enables users to define an axiom by using templates given by the tool with simple mouse operations. Axioms are translated into Prolog. The inference engine is based on Prolog and OKBC protocol[7] to make it implementation-independent.

OntoEdit

OntoEdit is software licensed ontology editor developed by Ontoprise [27]. It is an ontology editor that has been developed keeping five main objectives in mind [34]: ease of use, methodology guided development of ontologies, ontology development with help of inferencing, development of ontology axioms and extensibility through plug-in structure. Like most of other tools, OntoEdit employs the client/server architecture where ontologies are managed in a server and multiple clients access and modify. It is interoperable with other ontology tools like OntoAnnotate, OntoBroker, Semantic and Miner.

OntoEdit provides ontology storage in DBMS format, it support import from XML(S), OWL, Excel, RDF(S), Oracle, MSSQL, and MySQL and export to OWL, RDF(S), SPARQL, F-logic, and Excel. OntoEdit has built-in inference services. Another unique feature of this phase is that collaborative evaluation is also supported by introducing the name space so that the inference engine can process each of test sets given by multiple users. OntoEdit employs F-Logic as its inference engine. It is used to process axioms in the refinement and evaluation phases. Especially, it plays an important role in the evaluation phase.

[7] http://www.ai.sri.com/~okbc

2.5.3 Ontology Languages

Ontology languages allow users to write explicit, formal conceptualizations of domains models. The main requirements of ontology languages are [35]: a well-defined syntax, a well-defined semantics, efficient reasoning support, sufficient expressive power and convenience of expression. Several ontology languages have been developed during the last few years [36]. Some of them are based on XML syntax, such as Ontology Exchange Language (XOL), SHOE, Ontology Markup Language (OML), and Web Ontology Working Group of W3C[8] (WWW Consortium) identified a number of characteristic use-cases for ontologies on the Web like XMLS, RDF, RDF Schema, OWL, *etc.*

XML and XML Schema

XML is the universal format for structured documents and data on the Web, proposed by the W3C [27]. The main contribution of XML is that, it provides a common and communicable syntax for web documents. XML itself is not an ontology language, but XML-Schemas is, which define the structure, constraints and the semantics of XML documents to specify ontology. XML is a meta-language used to define other languages. It describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them [27]. XML defines neither the tags nor grammar, which makes it completely extensible [27]. It only requires that document must be well-formed in a tree structure, so it could be parsed by standard XML tools. In addition, the document can be structured to be valid. A valid document is one that conforms to its XML Schema, which defines grammar and tag set for specific XML formatting.

XML Schema enables common, well-defined and easy processable syntax, but nothing about semantics of data it describes. That means some standard must be needed on top of XML that describe semantics of data. The first step in that direction is Resource Description Framework (RDF), a general model in metadata layer and Resource Description Framework Schema (RDFS), language at schema layer.

RDF and RDF Schema

[8] https://www.w3.org/

RDF is a framework for metadata description developed by W3C [28]. The RDF data model defines a simple model for describing interrelationships among resources in terms of named properties and values. Resources are described in terms of properties and property values using RDF statements. RDF statements are represented as triples, consisting of a subject, predicate and object (S, P, O) [37]. RDF is written in XML and uses URIs (Unique Resource Identifiers) to identify resources.

RDF Schema, along with RDF provides basic capabilities for describing vocabularies that describe resources through other important features [28]. RDF schema has its own built-in classes and meta-classes by which users can define any class and relation. rdfs:Resource and its two subclasses: rdfs:Class and rdfs:Property are the key meta-classes. Every ordinary class defined in RDF Schema is an instance of rdfs:Class. In the same way, every property and relation defined in RDF Schema is an instance of rdfs:Property. The major role of RDF schema is to constrain the instance to which a tag is attached. On the other hand, the major roles of RDF schema include giving tags with definition and their taxonomy to RDF, though it also specifies constraints of the possible values of the triplet. XML is usable without XML schema, RDF is useless without RDF schema.

Web Ontology Language

The Web Ontology Language is a semantic markup language for publishing and sharing ontologies on the World Wide Web [35, 37]. There are three species of OWL: OWL Full, OWL DL and OWL Lite. OWL Full uses all the OWL languages primitives, it is fully upward compatible with RDF, both syntactically and semantically. Any legal RDF document is also a legal OWL Full document. OWL DL is a sublanguage of OWL Full which restricts the way in which the constructors from OWL and RDF can be used. It permits efficient reasoning support, but it loose full compatibility with RDF. OWL Lite excludes enumerated classes, disjointness statements and arbitrary cardinality among others. It is easier to grasp for users and easier to implement for tool builders.

OWL is the proposed standard for Web ontologies [37]. It allows us to describe the semantics of knowledge in a machine-accessible way. OWL builds upon RDF and RDF Schema. In OWL RDF syntax, instances and most RDFS modeling primitives are used. Formal semantics and reasoning support is provided through the mapping of OWL on logics. Predicate logic and description logics have been used for this purpose.

Simple HTML Ontology Extensions

SHOE[9] is an HTML-based knowledge representation language. SHOE is a superset of HTML which adds the tags necessary to embed arbitrary semantic data into web pages [36]. SHOE tags are divided into two categories [38]. First, tags are for constructing ontologies. SHOE ontologies are sets of rules which define what kind of assertions SHOE documents can make and what these assertions mean. Secondly, there are tags for annotating SHOE documents to subscribe to one or more ontologies, declare data entities, and make assertions about those entities under the rules prescribed by the ontologies. Because SHOE exists in a distributed environment with little central control, SHOE treats assertions as claims being made by specific instances instead of facts together and intern as generally-recognized truth. SHOE's syntax is a properly-compliant application extension of HTML, almost identical to XML syntax. However, while SHOE's chief application is the annotation of web documents, designed for more general distributed knowledge and distributed agent issues.

HTML function is for displaying data for humans to read. The knowledge on a web page is in a human-readable language (usually English), laid out with tables, graphics and frames in ways that we as humans comprehend visually. SHOE was designed with the needs of the web in mind [37]. It has limited semantics to make it possible to handle large amounts of data. However, simple database semantics are not enough for web data; SHOE provides true knowledge base semantics. It has a variety of mechanisms that try to deal with the fact that the data out there is distributed and under no one's total control.

Knowledge Interchange Format

KIF is a language designed for use in the interchange of knowledge among disparate computer systems [36, 37]. KIF was created to serve as syntax for first-order logic that is easy for computers to process. It is based semantically on predicate logic and syntactically on LISP. It allows

[9] http://www.cs.umd.edu/projects/plus/SHOE/spec.html

representing arbitrary sentences in the first order predicate logic. This language was defined within the Ontolingua project that provides languages, a cooperative ontology builder that allows exporting ontologies to various formalisms. When KIF is used, one usually implements representation formalism in KIF and uses this implementation for representation of particular ontology or knowledge. These ontologies can then be exported to other formalisms, such as Prolog. Even when KIF was primarily intended as Interlingua, it is currently used for encoding knowledge directly. Other formats may be exported from KIF definition automatically.

KIF is also not intended to be an internal representation for knowledge within computer programs or within closely related sets of programs. Typically, when a program reads a knowledge base in KIF, it converts the data into its own internal form (specialized pointer structures, arrays, *etc.*). All computation is done using these internal forms. When the program needs to communicate with another program, it maps its internal data structures into KIF. One inconvenience of this language is its computational complexity is too high [37]. Its expressivity is based on a version of first order predicate calculus, with extensions to support non monotonic reasoning and definitions. KIF has declarative semantics; this means that it is possible to understand the meaning of expressions in the language without the intermediation of any interpreter.

2.6 Event Modeling Approaches

A number of event modeling approaches have been proposed by different researchers. In general the approaches can be categorized as probabilistic, atomic, structural, generic, ontology based event model, *etc.*

2.6.1 Probabilistic Event Model

News event probabilistic model is proposed for Retrospective news Event Detection (RED) task in Topic Detection and Tracking (TDT) [39, 40]. Retrospective news event detection is defined as the discovery of previously unidentified events in historical news corpus [40]. It focuses on the utilization of both the contents and time information of news articles. In the model, news articles are represented by four kinds of information: who (persons), when (time), where (locations) and what (keywords), because news reports are always aroused by news events, a news event is modeled by mixture of three unigram models for persons, locations and keywords and one Gaussian Mixture Model (GMM) model for timestamps [40].

Historical events are a good supplement for linked data as it involves persons, places and other entities and it combine different entity types and add a historical component [40]. Concepts, semantic relations, facts and descriptions are used as a resource for several research areas like NLP, IR, IE and ontology building.

Overall, Wikipedia is a good source of event, because it is global resource containing important event triggers and participating entities in the world, contributors add new article and entities. A major challenge facing event detection from Wikipedia article is to separate the mundane and polluted information from interesting real world events. In practice, highly scalable and efficient approaches are required for handling and processing a large amount of data (especially for real time event detection). Using Wikipedia for retrospective event detection provide a good coverage for event extraction.

2.6.2 Generic Event Model

Generic event modeling was initially designed for event-centric multimedia data management components**.** Westermann and Jain, [42] proposed a generic event model E for event-centric multimedia data management applications. The author opted agile two step approach using object oriented design. E provides rich event descriptors using attribute values of arbitrary complexity, simple tags, and reference to concepts, media and other sensor data. The model offers genericity and extensibility to others eChronicle applications and it is also able to capture temporal aspect, spatial aspect, information aspect, experiential aspect, structural aspect and causal aspect of events.

The semantics of a constituent are denoted by its type, which is given by a Uniform Resource Identifier (URI) [43]. E leaves the definition of suitable types to applications, focusing on the modeling of events, however, E does not cover such type definitions and defers that to an event schema language. The type of each constituent is further augmented by the schema which is part of, again identified by a URI.

Minale A. and *et al.* [44] proposed a generic Multimedia Representation Space Model (MRSM), designed for multimedia data and multimedia-based event representation, in order to allow event detection and identification based on multimedia collective knowledge. MRSM provides a means

of extracting, representing, and linking events from unstructured and heterogeneous multimedia data without any prior knowledge about event-related clues.

2.6.3 Ontology based Event Modeling

Various ontology based event models have been proposed for representing real world events. The following are examples of ontology based event model: Event Ontology (EO[10]) [28], Linking Open Descriptions of Events (LODE[11]) [45], F-Model (F[12]) [46], Simple Event Model (SEM[13]) [48], EventsML-G2[14], NOEM [4] and others. Some more general models for semantic data organization also include event models like CIDOC-CRM[15] and the ABC[16]ontology [30, 47]. These event models differ significantly since they were created for diverse purposes and show different design choices.

News Ontology Event Model

The goal of designing NOEM is to provide a basic vocabulary for semantic annotation of event 5W1Hs in news stories [4]. The model captures temporal, spatial, information, experiential, structural and causal aspect of events. The model is able to cover information of events in three levels: event information, event relation and event media. The main concepts, properties of NOEM and how they are used to represent event semantic elements, is shown in Figure 2.1 [4]. *Happening* is the superclass of all types of eventuality. It has four subclasses: *Event* denotes dynamic event, *Situation* denotes static status, *Action* denotes an activity and *Constellation* denotes an event set.

[10] http://motools.sf.net/event/event.html

[11] http://linkedevents.org/ontology/

[12] http://isweb.uni-koblenz.de/eventmodel/

[13] http://semanticweb.cs.vu.nl/2009/11/sem/

[14] http://www.iptc.org/EventsML/

[15] http://cidoc.ics.forth.gr/official_release_cidoc.html

[16] http://metadata.net/harmoy/ABC/ABC.owl

Properties *hasCause* and *hasEffect* of *Happening* represent cause-effect connections between events.

Event is a concept denotes dynamic happening [4]. An event always has a type which can be used to specify *what*. *Situation* describes static status preceding or following an event. A property *precedes* and *follows* can be used to represent *how*. *Constellation* is a set of happenings with some relations among them, e.g., cause-effect and core-peripheral. It describes a complete happening caused by a key event and developed by sub-events. *Agent* is a concept to represent *who* and *whom*. It is the superclass of *Group* and *Person. Time* apparently represents *when*. *Place* represents *where. Document* is the media aspect of an event. It has an URI (Universal Resource Identifier) refers to a news article. *Topic* is a concept in document level. It is related to category of a news story, for example, Sports, Law, Politics, and so on.

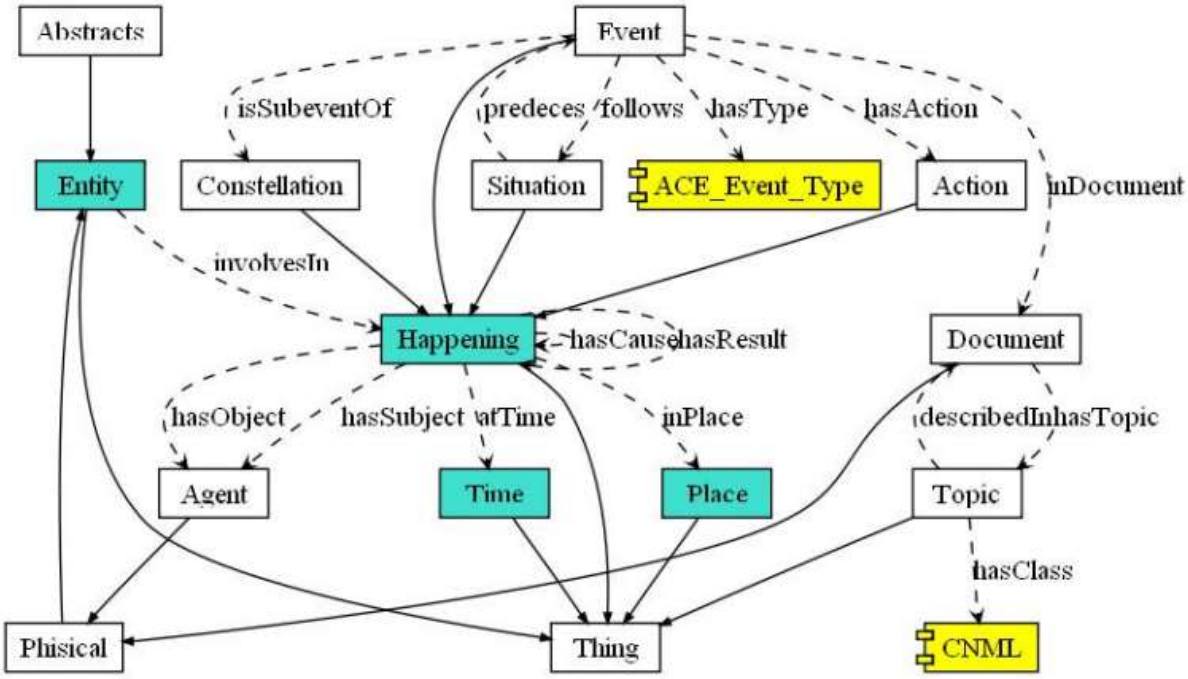

Figure 2.1: News Ontology Event Model

This model uses ACE-Event-Type. The task of ACE is limited to detection of eight specific event types which are: Life, Movement, Transaction, Business, Conflict, Contact, Personnel, and Justice. Each type has one to 13 subtypes so that each event is assigned to one main event type and one subtype of it. The limitation to these event types is the main obstacle why ACE Event Type cannot

be used in our setting directly, because we can identify more event types. Information such as news types, resource locators, or indexes corresponding to specific document that support the given event are modeled. CNML (Chinese News Markup Language) is imported to represent a news article's topic so that it can connect an event to its category in document level.

Simple Event Model

Simple Event Model is created to model events in various domains, without making assumptions about the domain-specific vocabularies used [48]. SEM is designed with a minimum of semantic commitment to guarantee maximal interoperability. Adopt principle of using external vocabularies. In SEM there is no cardinality restriction, as shown in Figure 2.2, SEM's classes are divided in three groups: core classes, types, and constraints. There are four core classes: sem:Event (what happens), sem:Actor (who or what participated), sem:Place (where), sem:Time (when). SEM's properties are divided in three kinds: sem:eventProperty, sem:type properties and a few miscellaneous properties like sem:accordingTo and sem:hasTimeStamps subproperties. The sem:eventProperty relates sem:Events to other individuals. A sem:type relates individuals of the sem:Coreclass to individuals of sem:Type.

SEM is characterized as a domain independent, class based event model of average size (in number of classes and properties), that contains only a few constraints [48]. SEM gathers the elements that give a light-weight description of events, but without importing strong semantic definitions that easily lead to inconsistency.

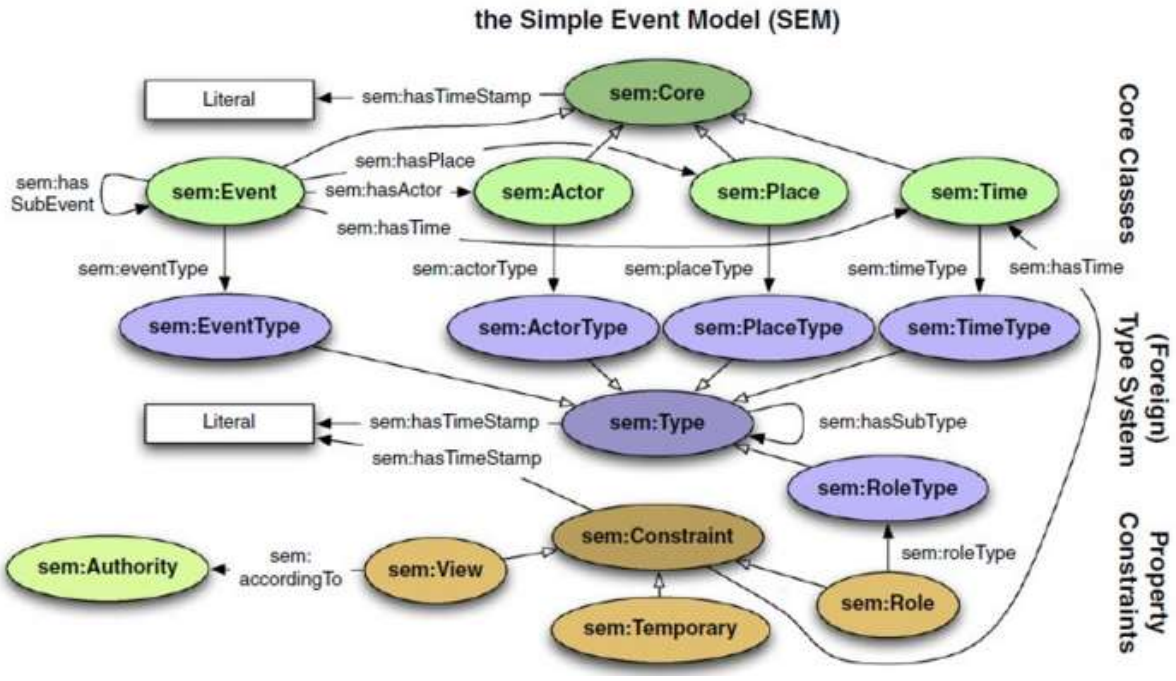

Figure 2.2: Simple Event Model Constructs[17]

Event Ontology

Event ontology is developed in the Centre for Digital Music in Queen Mary University of London [28]. It defines one main event concept. An event may have location, time, active agents, factors and products as depicted in Figure 2.3. Event Ontology follows a very simple design and consists of three classes Event, Factor and Product. It contains seventeen properties that relate events to property. This ontology is useful in a wide range of context, due to its simplicity and usability from talks in a conference, to description of a concert. EO defines a minimal event, and relies on vocabularies defined externally to refine the knowledge expressed. Roles, types, views and temporary are not defined in EO. Place, Time and Agent are defined via range restrictions on EO's properties. The explicit linking to vocabularies brings EO its richness, but also constrains the possible values for these properties. EO is domain independent, property-based model with few classes and constraints.

[17] http://semanticweb.cs.vu.nl/2009/11/sem/

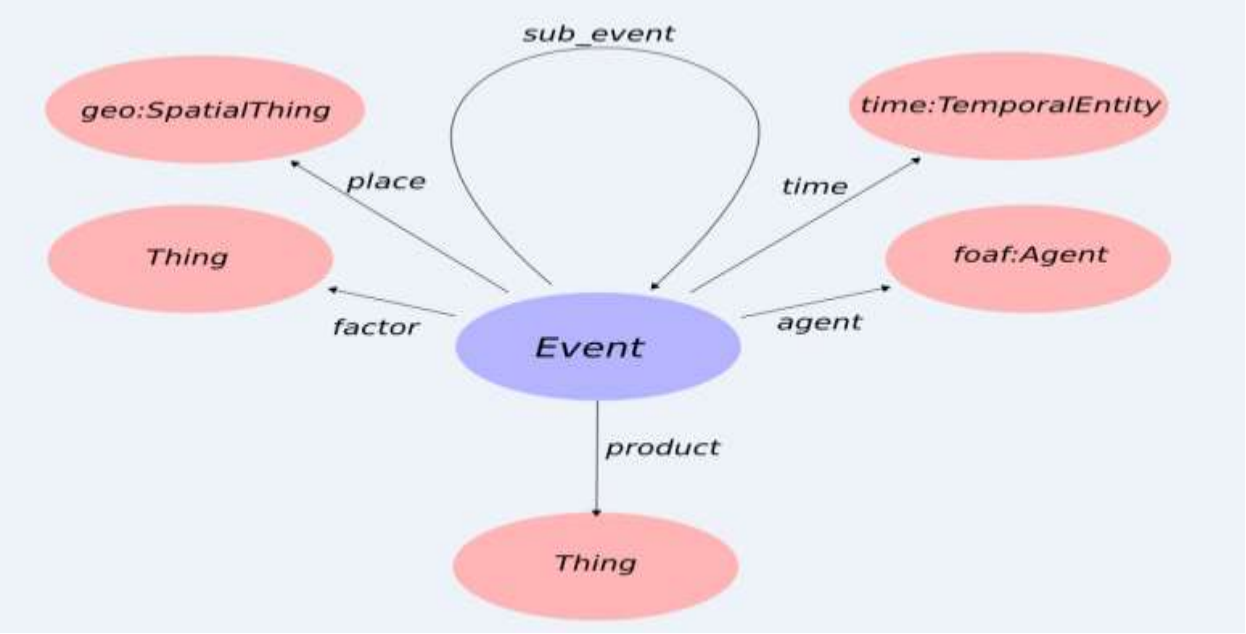

Figure 2.3: Event Ontology Model[18]

Linking Open Descriptions of Events

LODE aims at a minimal modeling of events and developed to represent historic and news events [45, 48]. It contains one class (Event) and six properties (lode: atTime, lode: circa, lode: inSpace, lode: atPlace, lode: involved, and lode: involvedAgent). Both class and properties are formally mapped to other event models like the CIDOC-CRM, EO and DUL by the use of owl:sameAs and rdfs:subPropertyOf. LODE is domain independent, property based model with few classes, some restrictions and a higher level of formalization.

A related problem is the question of event identification. An event is identified with a single textual description. They don't attempt to map multiple textual descriptions to the same event identifier. The reason for this is that it is not clear when two textual descriptions to be of the same event. If we consider (as many contemporary philosophers of history do) events to be linguistic phenomena rather than objectively existing in the past, then there is no basis for arguing that two textual descriptions of an event refer to the same thing.

[18] http://motools.sourceforge.net/event/event.html

F-Model

F-Model is formal representation of events that allows for capturing and representing human experience. This model is based on the foundational ontology DOLCE+DnS Ultralite (DUL) and provides comprehensive support to represent time and space, objects and persons, as well as causal and correlative relationships between events [46]. It provides additional properties, classes, part hood relation, casual relations and correlation between events. It has ability to represent multiple interpretations for the same event.

DUL defines the class DUL:Event next to the disjoint upper classes DUL:Object, DUL:Abstract, and DUL:Quality [46]. The Event class defines from the formal definition in DOLCE as an entity that exists in time. The class object stands for entities that exist in space, and abstract things like social and cognitive entities. A *Quality* is a characteristic of an object or an event. It has a value that is represented as a point or area in some *Abstract*. *Abstract* refer to the generic abstracts already defined in DUL such as the regions DUL:TimeInterval, DUL:SpatioTempor alRegion, and DUL:SpaceRegion. F-Model is domain independent, class-based model with complex scope, some restrictions and a higher level of formalization.

2.6.4 Comparison among different Event Modeling Approaches

Event models can be typed on basis of four main designs: domain specificity, classes or property centered, size and level of formalization. Table 2.1 shows the comparison among different ontology based event model.

Table 2.1: Comparison among different ontology based event model

Event model	Domain	Focus	Scope	Level of formalization	Purpose
EO	Independent	Property-based	Minimal	Higher	Music events
NOEM	Independent	Property-based	Minimal	Higher	News Events
LODE	Independent	Property-based	Minimal	Higher	Historic and news events
SEM	Independent	Class-based	Minimal	Average	---
F-Model	Independent	Class-based	Complex	Higher	Capturing human experiences
CIDOC-CRM	Dependent	Class-based	Complex	Lower	Museums, Libraries, and Archives artifacts

2.7 Tasks of Event Modeling

The task of event modeling requires, identification of event trigger, recognition of event arguments, finding event properties and annotation of extracted event as a metadata and storage [49, 50, 51]. Ahn [52] and Sharma *et al* [53] break down the task of extracting events into a series of classification sub-tasks. The first is anchor identification that is finding event anchors (the basis for event mentions). The second task is argument identification that is determining which entity mentions, and values of each event mention, attribute assignment that is determining the values of the modality, polarity, genericity, and tense attributes for each event mention. The third task is event reference, determining which event mentions refer to the same event. Each subtask is handled by a machine-learned classifier and highly interdependent.

2.7.1 Event Detection Techniques

Event detection is responsible for finding the most informative terms which describe the event mention [7, 54]. According to Aratefeh and Khreich [5], taxonomy of event detection is classified according to the type of event, detection task and detection methods is presented. Depending on the available information on the event of interest, event detection can be classified into specified (planned) and unspecified (unplanned) techniques [5]. Unspecified events of interest are typically driven by emerging events, breaking news, and general topics that attract the attention of users. Unspecified events are typically detected by exploiting the temporal patterns or signal of streams. Specified event detection includes known or planned social events. These events could be partially or fully specified with the related content or metadata information such as location, time, agendas, and participants.

According to the detection task and target application, classified into Retrospective Event Detection (RED) and New Event Detection (NED) techniques [55]. Retrospective detection entails the discovery of previously unidentified events in a chronologically-ordered accumulation of stories. The other is online new event detection system has two online modes of operation [55, 56]: immediate and delayed. In immediate mode, a strict real-time application is assumed, and the system indicates whether the current document contains or does not contain discussion of a new event before looking at the next document. In delayed mode, classification decisions deferred for a pre-specified time interval. For example, the system could collect news throughout the day and provide the user with new events at the end of the day.

Furthermore, depending on the detection method, the presented techniques are also categorized into supervised and unsupervised (combination of both) techniques. In addition, the main event detection methods, the feature representations, which are divided into general and domain-specific features.

2.7.2 Event Extraction

Event extraction is a concept that is crucial in event management, intelligence gathering, and decision making [6, 10]. One of the main objectives of event modeling is the ability to automatically detect news articles that contain precise information about events. To improve such unusual event identification, present tense verbs, popular event nouns and adjectives that describe

events as they take place are considered typical features [6]. A bag of words model uses a dictionary of trigger words to detect and characterize events which are manually labeled by experts from several management departments: traffic control, crisis, emergency departments, and others [57, 58].

Event triggers are recognized using conditional probability method [59]. A sample POS tagged document contains the text fragments and explain occurrence of the type of event to be extracted and the POS tagged text from which event is to be extracted. Bayes theorem is applied to find the probability of occurrence of the words of input text in the sample documents as specified in equation 2.1.

$$\mathrm{P}(\mathrm{V}|\mathrm{V1},\mathrm{V2},\ldots,\mathrm{Vn}) = \frac{\mathrm{P}(\mathrm{V1},\mathrm{V2},\ldots,\mathrm{Vn}|\mathrm{V})\,P(V)}{\mathrm{P}(\mathrm{V1},\mathrm{V2},\ldots,\mathrm{Vn})} \qquad \ldots\ldots (2.1)$$

where V is the word found in input text and V1, V2, ... Vn are words found in sample documents. Words with high probability are selected as event triggers. As defined in ACE [60] guidelines and Sangeetha *et al* [59], if the combinations of two words are of POS verb and noun, noun is chosen as the trigger. If the combinations of three words are of POS verb, any POS, and adjective respectively, the word with POS adjective is chosen as the trigger. If the combinations of two words are of POS verb and particle, both the words with POS verb and particle are chosen as the trigger. A single word is generated as an event trigger if the entire event is specified in a single sentence. Multiple words are generated as event trigger if the entire event spans more than one sentence or specified in a single sentence using conjunctions.

Method of extracting event trigger word based on trigger word table [57, 60]. It follows two processes: construct trigger word set; calculate weight value. First, do sentence segmentation and mark POS by using segmentation tool, then filter out part of words and phrases in the word collection formed by the segmentation. This action can narrow the scope of candidate trigger word set, and then describe the set in the format of equation 2.2

$$W = \{(w1, score1),(w2,score2),\ldots, (wk, score_k)\} \qquad \ldots\ldots (2.2)$$

where w stands for candidate trigger word, score stands for the word's weight value. Weight is computed using TF and Inverse Document Frequency (IDF). TF(term frequency) refers to the number of occurrences of the given word in the document. For word wi, its term frequency can be calculated by using equation 2.3.

$$TF(wi) = \frac{ni}{N} \quad \text{(2.3)}$$

where ni is the number of wi that appears in document, N is the total number of all candidate trigger word in the document. IDF is a measure of a word's general importance; it can be determined using equation 2.4.

$$\text{IDF(wi)} = 1 + \log_2(trigger\ word's\ weight\ value) \quad \text{(2.4)}$$

Then threshold is set, and filter some candidate word whose weight value is less than threshold. Normalized term frequency-inverse document frequency (tf-idf) is the product of TF and IDF computed using equation 2.5.

$$Score(wi) = TF(wi) * IDF\ (wi) \quad \text{(2.5)}$$

Event recognition is defined to be the task of associating incoming stories with events known to the system [39, 55]. In the recognition task, a target event is given, and each successive story must be classified as to whether or not it discusses the target event [39]. The objective of event recognition is to correctly classify all of the subsequent stories. Thus each target event is defined by a list of stories that discuss it. New event recognition [3]: aims to recognize the first story for a new event that had not been discussed before. Each news article is processed in sequence, and a decision is made whether or not a new event is discussed in the story, before processing any subsequent stories. A decision is made after each story is processed. The first story to discuss an event should be flagged YES. If the story doesn't discuss any new events, then it should be flagged NO.

2.7.3 Knowledge Representation

Knowledge representation is an old field in Artificial intelligence and has provided numerous models from frames to recent variants of description logics, RDFS and OWL [35]. Objects, properties, categories and relations between objects, situations, events, states and time, causes and effects are the things that artificial intelligence needs to represents. Knowledge representation techniques are divided in to two major categories that are declarative representation and procedural representation [26]. The declarative representation techniques are used to represents objects, facts, relations, whereas the procedural representation are used to represent the action performed by the objects. The propositional logic, predicate logic, semantic network are the declarative knowledge

representation techniques and script, conceptual dependency are procedural knowledge representation techniques.

DBpedia, YAGO [62], Freebase and Google knowledge graph are well-known examples of knowledgebase constructed using information extraction techniques. Information extraction techniques can extend existing knowledgebase; in turn knowledgebase can also improve information extraction.

A semantic network is widely used knowledge representation technique [26]. Semantic network is a knowledge representation technique in which the relationship between class and objects are represented by the connection or link between objects or class of objects. Thus the notion of semantic extends the current service by standardizing its semantics, adding richer and more meaningful interaction among users and machines. Another form of knowledgebase construction is ontology population from unstructured documents. W.Wang and D. Zhao, [4] populate ontology by using a predefined template; they automatically generate an OWL file for extracted event facts and import them into the ontology to build an event knowledge base. Event based media repository is discussed in [63].

Word vectors encode semantic meaning and capture many different degrees of similarity [64]. Word2vec is a tool developed by Mikolov.et al. [64, 65]. Word2vec captures domain similarity while other more dependency-based approaches capture functional similarity. Word2vec provides an efficient implementation of continuous bag-of-words and skip-gram architectures for computing vector representations of words [66, 67]. These representations can be subsequently used in many natural language processing applications and other research areas like word embedding, mapping to semantic representation or syntactical space.

2.8 Summary

Most of the time events are associated with news broadcast on Web, social media, television, radio and others. Event modeling needs understanding of various disciplines such as natural language processing, knowledge representation, ontology engineering, and *etc.* NLP tools plays significant roles in event modeling because event extraction requires different NLP tools like POS tagger, named entity recognition, morphological analyzer, *etc.* are necessary for successful implementation of new event detection and recognition.

Event detection can be done offline mode for retrospective events or online mode, immediate or delayed new event detection. Task of event modeling involves: key event identification; semantic event element extraction and knowledgebase representation. Event extraction requires identification of event trigger, recognition of event arguments, finding out event properties and definition of extracted for knowledge base construction.

There are different approaches of event modeling like probabilistic, atomic, structural and generic which don't require the notion of ontology. Other ontology based event modeling approaches such as NOEM, EO, LODE, F-Model, SEM *etc.* can be used for event modeling. These ontology based event modeling approaches differ in domain specificity, size, purpose and level of formalization.

Chapter Three

Related Work

3.1 Introduction

A number of research works have been conducted on event extraction from various sources for different target applications. In this chapter we present the review of papers that are particularly related to our work. Our review presents the approach, data sets used, results obtained, weaknesses, strengths, lesson learned and subjects that make our research different from those works is identified.

3.2 Event Extraction from Wikipedia

Fabian, *et al.* [62] proposed YAGO, a large event ontology constructed from Wikipedia and WordNet. Event facts are automatically extracted from Wikipedia and unified with WordNet, using combination of rule based and heuristic methods. The authors have downloaded the English version of Wikipedia in January 2007, which comprised 1.6 million articles and each Wikipedia article is a single Web page and usually describes a single topic. They used version 2.1, WordNet contains 81426 synsets for 117097 unique nouns. WordNet distinguishes between words as literally appearing in texts and the actual senses of the words. In addition, it provides relations between synsets such as hypernymy, hyponymy i.e., the relation between a sub-concept and a super-concept. Wordnet also includes other types of words like verbs and adjectives, but they considered only nouns for their work, even though verbs and adjectives have capability to show the mention of events.

YAGO is based on a logically clean model, which is decidable, extensible, and compatible with RDFS, and available in different export formats, including plain text, XML, RDFS and SQL database formats [62]. YAGO is tailored to Wikipedia and WordNet, but it comes with a multitude of interconnected relations, and opens the door to numerous new challenges. All YAGO's facts are tagged with a confidence value between 0 and 1, and facts with their empirical confidence estimation of 0.90 to 0.98 are selected. Their evaluation result shows, YAGO has near-human accuracy around 95%. The resulting knowledge base is a major step beyond WordNet: in quality by adding knowledge about individuals like persons, organizations, products, *etc.* with their semantic relationships and in quantity by increasing the number of facts. The overall number of ontological facts is about 5 million. This number is completed by the respective witness facts and

approximately 40 million context facts. The size or number of entities in YAGO is: 14 relations, 149162 classes and 907462 individuals.

Another works done by James, *et al.* [41] uses Wikipedia as a target source of event. Wikipedia contains historical events of different granularity in lists for centuries, years, months and on a daily basis. The focus of [41], is to extract historical events from Wikipedia articles that are available for about 2,500 years for different languages. According to [41] events can be identified either by being of the DBpedia[19] ontology type event or containing a date attribute (Date, StartDate, *etc.*). Accordingly events have a date at the beginning, followed by a short text describing the event with links to other Wikipedia articles. Their event extraction is based on semantic parsing from Wikipedia text and converted them into the LODE event model. They have extracted historical events in different languages like German (36063), English (32943), Spanish (18436), Romanian (9745), Italian (6918), Portuguese (6461), Catalan (6442), Turkish (3084) and Indonesian (1720) with a temporal coverage from 300 BC to 2013. Totally, these results in 121812 events with 325693 links to other Wikipedia articles, which means an average of about 2.7 links per event.

For the individual languages their algorithm achieved the following extraction quotients: German (98,97%), Spanish (94,12%), Turkish (91,87%), Portuguese (91,74%), English (86,20%), Catalan (85,69%), Indonesian (80,19%), Italian (75,81%) and Romanian (74,73%). Historical events are a good supplement for linked data as it involves persons, places and other entities available in DBpedia and give background information for certain phenomena. They have parsed and processed historical events on a yearly basis for different language versions and make them available by a Web API, SPARQL endpoint, Linked Data Interface and in a timeline application. They have applied retrospective event detection; which is used to extract event from large chronological accumulation of articles. This approach is not suitable for online event extraction, because it require immediate (real time) event detection. In addition to this, they have considered existence of datetime as a parameter for event detection, but all resources containing datetime is not considered as event.

[19] DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web.

3.3 Event Extraction from News Article

Wang *et al*. [1] proposed a framework that extracts event semantic elements (5W1H) from Chinese news for event ontology population. Their approach divide event extraction task into three steps: identify key events of news stories, extract 5W1H semantic information from the key events, and represent the extracted semantic information of events to build event knowledge base. They have designed News Ontology Event Model (NOEM) to represent 5W1H semantic elements of an event and relations among events. The extracted semantic event elements are first represented as RDF (Resource Description Framework) triples[20], and then automatically imported into NOEM as instances. As a dataset they used news stories and manually annotate them with title type, topic sentences and key event 5W elements. As a result they construct three datasets: ACE05, it contains 182 XinHua newswire articles which come from ACE2005 Chinese training corpus, BJRB, which contains 543 news stories selected from March 1-10, 2009 Beijing Daily, and XAWB, which contains 808 news stories selected from June 1-10, 2009 Xi'an Evening News.

Term frequency-inverse document frequency (tf-idf) and PageRank based methods were used to extract topic words from a news story and identify whether the news headline is informative or not [1]. Their evaluation result shows that the tf-idf based method has a better classification accuracy than the PageRank based method. They used combination different features to weight sentences and choose the top score one as topic sentence. Their experimental result shows that title and title classification are both important to find a topic sentence, the first sentence is more important than the title, and the title is above other features. For the 5W tuple evaluation, they used string similarity measure to compute the similarity between the extracted Time, Location, Subject, Predictive, Object elements of the 5W-tuple <T, L, S, P, O> and manually annotated results. Their primitive evaluation results show *when* and *where* are better than *who* and *whom.* It makes sense because time and location are comparatively easier to identify. The *who* and *whom* are closely related to *what*, so the accumulated errors lead to a lower precision. The result also shows that their algorithm tends to be affected by the document length.

Jakub *et al* [68) proposed a NEXUS (News cluster Event eXtraction Using language Structures), which uses Ontology of Politically Motivated Violent Events (PMVE). Before the NEXUS event

[20] RDF triples are used to group any two pieces of data and the link that connects them on the web.

extraction process can proceed, news articles are gathered by media monitoring software (EMM system), which delivers news clusters for each topic. Further, NEXUS selects security related events via application of keyword based heuristics. The extracted information about violent events and related entities such as people and organizations is mapped to domain ontology PMVE. The PMVE domain ontology consists of the concepts, the relationships and the properties that formally describe the PMVE as they are reported in the online news articles. The purpose of the ontology mapping process is to assign each violent event to its relevant concept in the ontology and obtain the instances for the knowledge base. Gathering information about violent events is an important task for better understanding conflicts and for developing early warning systems.

Initially, they have used 84 concepts covering events, places, people, groups and organizations in the ontology, 59 properties describe these concepts and 30 relationships connect them with each other. Evaluation is carried out on 26333 English language news articles grouped into 826 clusters from the period 24-28 October 2006. They implemented in nexus an ad-hoc text classification algorithm for detection of security-related news clusters based on frequency of patterns matched, number of articles and keywords present. The algorithm classified as security-related 47 events; for 39 of them the classification was correct. Regarding the identification of places, geo-location algorithm is precise at the level of country (95%), but due to some temporary technical problems relatively low accuracy (28%) at the level of city, town and village can be observed. Automatic event extraction from free text is an extremely relevant application for many areas such as risk assessment and early warning. The evaluation revealed that there is still space for improvement, especially in case of date detection and security-related events classification.

3.4 Event Extraction from Social Media

Axel *et al*. [69] proposed event recommender system for Twitter users. It identifies twitter activity co-located with previous events, and uses it to drive geographic recommendations via item-based collaborative filtering. They collect Twitter data from the twitter public streaming API, and stored on a hadoop file system, which is filtered down from high activity areas of their interest such as New York and Chicago. Lists of events are extracted from a web crawl of Eventbrite, a popular event organization website which contains a database of previous and upcoming events. These events are cross-referenced with geotagged twitter traffic to identify users who have attended these events. From the tweets that the user created or received or retweeted they mine the user's opinion

about the event using sentiment analysis. The intensity of user sentiment about an event is translated into a rating about the event.

Event recommendation presents a unique challenge because we cannot measure user interaction with the event until it has already occurred. In order to recommend events to users, they used item-based collaborative filtering to recommend upcoming events similar to ones previously attended by the user. Similarity between two different events is calculated based on event description, user participation and social media interaction during and about the event. This event recommendation can be applied to any other social media network user activity that has ties with geo-location and time of the activity. Their works explore only techniques to detect and recommend real world events for users based on their Twitter activity. However event semantic elements extraction and definition is uncovered. Higher result could be achieved during similarity measure, if the four dimension or elements of events extracted and considered individually, because with in fraction of second similar event at different time or location may happen.

Hila *et al.* [70] present a query-oriented solution for retrieving social media documents for planned events across different social media sites. Automatically identifying social media content associated with known events is a challenging problem due to the heterogeneous and noisy nature of the data. They mine event aggregation platforms to extract event features, which are often noisy or missing and they used these features to develop query formulation strategies for retrieving content associated with an event on different social media sites. User contributed Web data contains rich and diverse information about a variety of events in the physical world, such as shows, festivals, conferences and more. This information ranges from known event features (e.g., title, time, location) posted on event aggregation platforms (e.g., Last.fm events, EventBrite, Facebook events) to discussions and reactions related to events shared on different social media sites. Their approach for associating social media documents with planned events consists of two steps, precision-oriented queries for an event using its known context features and recall oriented queries, to retrieve additional documents for the event. Planned event dataset collected from records posted between May 13, 2011 and June 11, 2011 on four different event aggregation platforms: Last.fm events, Eventbrite, LinkedIn events, and Facebook events. They collected a total of 393 events, with 90 events from Last.fm, 94 events from Eventbrite, 130 events from LinkedIn, and 25 events from Facebook. For training they used a separate set of 329 event records, collected (in a similar fashion) between April 26 and May 11, 2011.

Overall, their evaluation showed that, query generation approaches can effectively retrieve relevant social media documents for planned events on multiple social media sites. The bulk of information from events is contributed by individuals through social media channels: on photo and video-sharing sites (e.g., Flickr, YouTube), as well as on social networking sites (e.g., Facebook, Twitter). This event-related information can appear in many forms, including status updates in anticipation of an event, photos and videos captured before, during and after the event, and messages containing post event reflections. Importantly, for known and upcoming events (e.g., concerts, conferences) revealing, structured information (e.g., title, description, time, and location) is often explicitly available on user-contributed event aggregation platforms (e.g., Last.fm events, Eventbrite, Facebook events). Generally, their techniques reveal important information from and about planned events as they are reflected through two hundreds of millions of users on social media sites.

3.5 Summary

The task of event extraction differs according to the type of event, detection task, detection and methods. Design and usage of source (Wikipedia, news article, social media or other) has its own impact on the event extraction process. A major challenges in event detection from these source is to separate the mundane and polluted information from interesting real world events. Event detection from such sources need to handle the noise data like spelling and grammar errors, length of text messages(maximum of 40 characters in twitter), improper sentence structure, frequent use of informal, irregular, abbreviated and mixed languages are common challenges, as a result some works uses media specific NLP tools. In practice, highly scalable and efficient approaches are required for handling and processing the increasingly large amount of data, especially for real time event detection.

Some works try to mine user's opinion using planned event of social media and use this information to recommend the upcoming event. User interaction before, during and after occurrence of event is an important indication for the happening of event, most works use only event trigger word to detect an event, but all article containing event trigger is not indicate the mention of event, while others consider only planned events published over event aggregation platforms. Here our concern is news article published over the web considering both planned and unplanned events. To address the challenge of automatically identifying unknown events,

considering the high rate of news broadcast, news shared over social streams and the dynamic nature of Web data, we proposed event extraction, and representation model from Amharic news article.

Moreover event modeling is language dependent; one should take language features and encoding techniques into consideration. Development of such language specific event modeling requires efficient NLP tools (Part of speech tagger, morphological analyzer, and named entity recognizer). Amharic is one of morphologically rich language, has its own character, numbering, way of writing and encoding. Thus language specific issues like normalization (character, and temporal), use of NLP tools should be considered. Beside these, extracting key event and semantic event elements from news article is not enough, because way of representation is also an issue. Here our concern is to use ontology and optimize the representative ability of event facts through semantically rich event representation model that can be used as a knowledge base for other event level semantic applications. Thus, the above mentioned related works can't address these issues for Amharic language and to our knowledge this work is the first attempt for Amharic language.

Chapter Four

Event Modeling from Amharic News Article

4.1 Introduction

In this chapter, we discuss the proposed event modeling from Amharic news article. First, we discuss event task definition and assumptions, then the proposed architecture of event modeling from Amharic news article and each components of the architecture along with techniques of implementation are presented in detail.

4.2 Event Task Definition and Assumption

Events are dynamic data structures that play key role in understanding real world eventuality, which are basically driven by the four 'W's' (what, who, when, and where) that is what happened, when and where did it happen, and who was involved. This natural progression of questions is a classic example of what one might ask about an event. Without any one of these questions, the news article telling about event would fall level and quickly become less attractive. For example, what is a news article about earth quake without a mention of location and time? Figure 4.1 shows the basic dimension of event in our assumption. Each of the four W's is necessary to tell full information about event. Thus we adapt the 4W concept in journalism[21] to represent semantic elements of events.

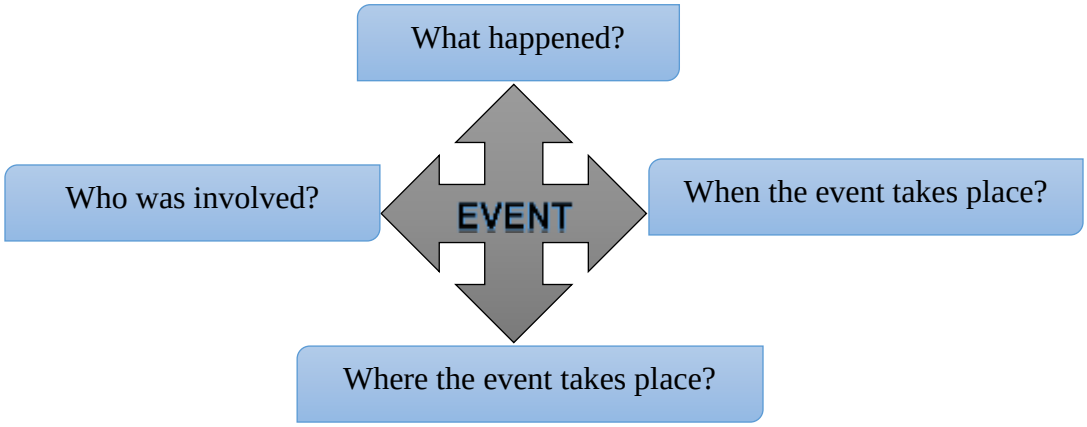

Figure 4.1: The Four Dimensions of Event

[21] http://iptc.org/standards/newsml-g2/

Events provide a natural way to explain complicated relations between people, places, actions and objects. Event centered modeling captures the dynamic aspects of an event along with related events and participating entities. The four dimensions of event presented in Figure 4.1 have ability to represent event semantic elements.

In order to capture and represent event facts, we defined the following terms and tasks mathematically. Event terms and tasks are defined according to assumptions made in this work.

Definition 1: [Event *E*]. We define an event E as a real world phenomenon that occurred at specific time *T*, involving participant *P*, and tied to a location *L* that published or posted to one of the public media available on the Web (News broadcast, social media). It is represented in four main attributes.

$$\boldsymbol{E} = (T, P, L, K) \quad (4.1)$$

Where,

T: temporal information describes the time **when** an event occurs (instant or interval).

P: entities participated or intended audience (at real time or after), to describe **who** was involved/attended,

L: spatial information (location or place) to describe the place **where** it took place,

K: textual description of the event, describe **what** is published,

Event Trigger: event trigger is word or phrase that most clearly express the mention of an event.

Event Extent: An event extent is a sentence within which a taggable event is described.

Entity: Person, Organization or Location contained or mentioned in the content of an article

Article: An instance of a news report

Event Repository: A database of identified events

Event Modeling: is the process of event identification, event elements extraction and event semantic elements representation.

When designing an event model that is supposed to meet the expectations in this work, several aspects need to be considered. In general the design should be simple and expressive model that

could be easy to understand both in terms of processes and topology. The following basic tasks are considered during the design of the proposed model.

- Preprocessing free Amharic news article and automatically identify news article containing the mention of events, in such away it is possible to address events from Amharic news article.
- Extracting event elements that provide rich means to capture elementary aspects like temporal, spatial, informational and participants of an event description.
- Representing the extracted event facts and their pairwise semantic relations, which together form an interconnected network of events.

4.3 The Architecture of Event Modeling from Amharic News Article

The proposed event modeling architecture consists of five major components: preprocessing, event trigger identification, event semantic elements extraction, classification and event representation. Figure 4.2 shows the architecture of event modeling from Amharic news article. Detail description and implementation algorithms for each component is discussed in following sections.

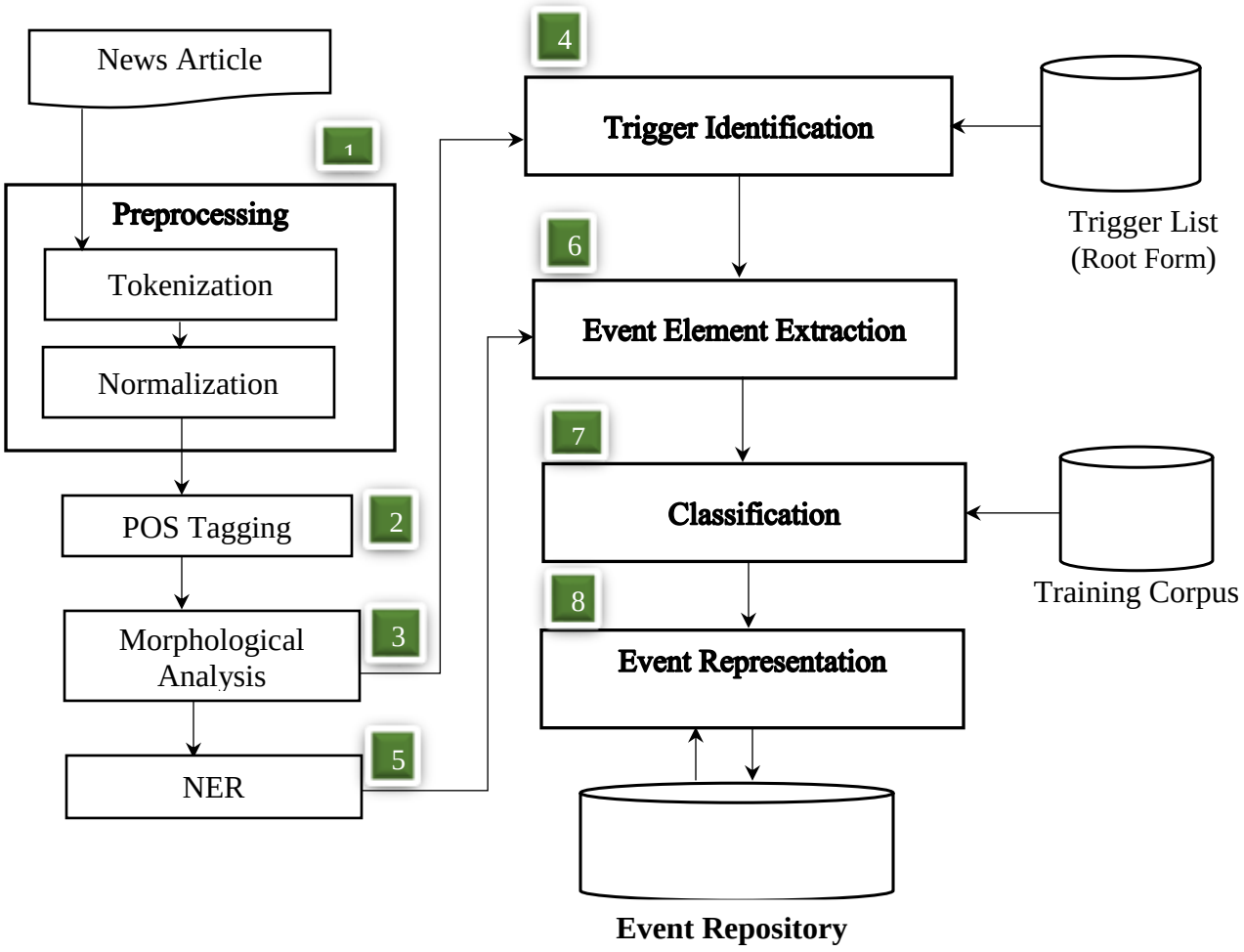

Figure 4.2: The Architecture of Event Modeling from Amharic News Article

4.4 Preprocessing

The main task of preprocessing module is to make the news article ready/soft for event extraction processes. As shown in Figure 4.2, preprocessing is the initial task for event extraction process. It accepts unstructured raw natural language texts as input for processing. In preprocessing language specific subcomponents like tokenization, normalization, and different linguistic components like part of speech tagger and morphological analysis and named entity recognizer are performed. The result of the preprocessing module is segmented news article into sentences and tokens, normalized (character and temporal expression), that is processed news article which is ready for trigger identification and other components. Each subcomponent is discussed in the next subsections in detail.

4.4.1 Tokenization

Naturally, before any real text processing is to be done, text needs to be segmented into linguistic units such as words and sentences. The incoming news article is plain text, we need to find where do sentence start and end, where are the words, whitespace, and punctuation marks. Since most written languages have punctuation marks which occur at sentence boundaries. As a sentence is one of the most important components in the natural language text for representation of interrelated information and for expressing event in news article. Algorithm 4.1 is used to tokenize news article into sentence level.

Algorithm 4.1: Sentence Level News Article Tokenizer Algorithm

```
Input: News Article(A)

Output: Tokenized news article into list of sentence

Begin

For every term in A

   Find punctuation marks :: or ! or ?
   If punctuation marks found
       Split sentence
   End if

End for

End
```

In practice, sentence and word tokenization cannot be performed successfully independent from one another. Since words and sentences identified at this stage are the fundamental units passed to further processing stages such as normalizer, part-of-speech tagger, trigger identification and event elements extraction components. Algorithm 4.2 tokenize news article into list of words by finding Amharic word separator.

Algorithm 4.2: Word Level News Article Tokenizer Algorithm

```
Input: News Article(A)

Output: Tokenized news article into list of words

Begin

For every term in A

   Find punctuation marks white space

     If punctuation marks found

       Split word

   End if

End for

End
```

4.4.2 Normalization

Normalization refers to the process of mapping the recognized expression to their specific standard format or measures. The normalization task is required because some information types do not conform to a standard format. This task is typically achieved through the use of conversion rules that produce a standard format. Amharic language specific normalization like character and temporal expression normalizations are discussed in this section.

Character Normalization

Character normalization component is responsible to normalize Amharic characters having same pronunciation and meaning into single character. Algorithm 4.3 iterates over news article to find redundant characters and replace with norm character. During normalization, the algorithm consider the seven orders redundant characters with corresponding norm orders.

Algorithm 4.3: Amharic Character Normalizer Algorithm

```
Input: Tokenized News Article(A)
Output: character normalized news article
Begin
For every character(C) in A
   if C == ሐ or C == ኀ or C ==  ኸ
      replace C with ሀ
      apply for the orders of C, considering the order of ሀ
   if C == ሠ
       replace C with ሰ
      apply for the orders of C, considering the order of ሰ
   if C == ዐ
       replace C with አ
       apply for the orders of C, considering the order of አ
   if C == ፀ
       replace C with ጸ
       apply for the orders of C, considering the order of ጸ
End for
End
```

Temporal Normalization

To resolve the ambiguity of temporal expressions, we have implemeted a rule based temporal expression normalizer. Most temporal expressions in news articles are incomplete and only implcitly anchored, often with respect to the dateline of the news article, wich refer to as the article's temporal anchor. The values of relatively simple temporal expressions like today(ዛሬ), yesterday (ትናንት), or tomorrow (ነገ) are computed with respect to this temporal anchor. As

indicated in Table 4.3, the rules for today simply assigns the reference, while the attachement for tomorrow and yesterday add a day and substract a day from the reference, respectively. Complex temporal expressions like a day after yesterday (ከትናንት በስትያ) are resolved one expression followed by other, resolve yesterday(ትናንት) first then apply the changes that the word after(በስትያ) makes on the date obtained. In Table 4.1 a sample temporal reference and for normalization rule are defined. These rules are captured only considering most frequent temporal expression in news article.

Table 4.1: Temporal Reference and Rules for Temporal Expression Normalizer

Temporal Reference	**Rules**
Today(ዛሬ)	day()/month()/year()
Yesterday(ትናንት)	day()-1/month()/year()
Tomorrow(ነገ)	day()+1/month()/year()
Next month(ቀጠይ ወር)	day()/month()+1/year()
num + year later (ከ num ኣመት በኋላ)	day()/month()/year() + num
A day after yesterday (ከትናንት በስትያ)	(day()+2/month()/year()

In some cases the references are estimated using the news article's date. Others refer to a date named before in the text that is being resolved. By default, the news article's date is used as a base referent temporal expression if it exits, if not, the system date is used.

4.5 Part of Speech Tagging

Part-of-speech tagger is one of basic text processing task. After news article is tokenized the next task is tagging the part of speech of determined phrases or tokens. The POS tagger receives tokenized news article as input and assigns POS tags to the words of that sentence that is output the lexical category of tokens such as nouns, verbs, adjectives, *etc.* For this we used a medium sized POS tagged corpus for Amharic consists of 1,065 news articles (approximately 210,000 tokens) collected from Walta Information Center [15].

We used part of speech for filtering candidate triggers in event trigger identification. Usually events are triggered by verbs and nouns. From the tagged data we can easily identify candidate triggers.

4.6 Morphological Analysis

Morphological analysis is the process of mapping various syntactic forms of a word into its canonical base form. The purpose of this process is to reduce the variation among event triggers, which reduce the sparseness for lexical representation of the text. Morphological analysis is a standard technique which is commonly used to create the bags-of-words features in many information extraction systems. Morphological analysis is applied to list of candidate event trigger extracted from news article and trigger list. To handle morphological variety of tokens we used manually constructed morphological analysis. It accepts words and reduces to its root word. The following are sample example that shows how morphological analyzer reduce various surface forms (derivation and inflection) of words into its root form. Morphological analyzer reduces various surface forms (ጉባኤ፣ የጉባኤዉን፣ ከጉባኤ፣ ገባኤን፣ . . .) into root form ጉባኤ.

Surface form	Morphological Analyzer	Reduced root form
አደጋ	አደጋ	
አደጋዎች	አደጋ-ዎች	
አደጋውን	አደጋ-ውን	አደጋ
አደጋን	አደጋ-ን	
ከአደጋዎች	ከ-አደጋ-ዎች	
. . .		
ጉባኤ	ጉባኤ	
የጉባኤዉን	የ-ጉባኤ-ዉን	ጉባኤ
ከጉባኤ	ከ-ጉባኤ	
ገባኤን	ገባኤ-ን	
. . .		
ከሰረ	ከሰረ	
ከሰረች	ከሰረ-ች	ክ-ስ-ር
ከሰሩ	ከሰርኡ [ከሰሩ]	
የከሰረ	የ-ከሰረ	
. . .		

4.7 Trigger Identification

Event is made up of event trigger and event elements, thus event extraction process start form preprocessing and followed by event mention (trigger) identification. The output of preprocessing

component is used as input for event trigger identification. Event trigger identification is process of identifying keyword/action words or phrases that can tell the mention of events. Event triggers are used to identify the mention of event in news article. For example, the following statement describes a traffic accident (ትራፊክ አደጋ) event.

“በአዲስ አበባ ነፀ ማዞሪያ ዛሬ ጠዋት የደረሰ ከፍተኛ የትራፊክ አደጋ የሶስት ሰዎች ህይወትን ሲቀጥፍ በርካቶችንም አቁስሏል።”

The description of event scenario depends on the level of abstraction. The above event description can be alternatively described as “በአዲስ አበባ ነፀ ማዞሪያ ዛሬ ጠዋት የደረሰ ከፍተኛ የትራፊክ አደጋ የሶስት ሰዎች ህይወትን ቀጠፈ” (highly specific), to “ከፍተኛ ትራፊክ አደጋ ደረሰ” (abstract).

According to rules of journalism and structure of news[22], a single article tells about one thing and most essential information about news article is found in news headline or at the first paragraph. In some cases a single news article might describe more than one event, but only one is a key event. Key event can be extracted by analyzing information content of news article using event trigger list and other local features.

Event trigger is a word or phrase starting an event, which is an important feature for recognizing the mention of event. From the above example “ትራፊክ አደጋ” (traffic accident)” is an event trigger. To identify event trigger we used a set of features of candidate trigger like part-of-speech of tagger, exact-match (if the trigger exactly match) and root-match (if the root of trigger match. Event trigger selection is done through filtering features with performance measures in mind and remove irrelevant features that might introduce greater computational cost. Such words are found through extensive analyze of textual features in order to select the best contributors to the task of event identification. For this task we have collected different event trigger words or phrase from the training corpus containing news article of different domains on the ground truth positive instances and sample list of identified event trigger words are presented in Appendix II.

[22] http://drkblake.com/six-rules-for-writing-a-straight-news-lead/

Table 4.2: Sample event extent and event trigger

#	**Event Extent**	**Event Trigger**
1	በሻሸመኔ ከተማ ትላንት ምሽት በደረሰ ድንገተኛ **የእሳት አደጋ** የሁለት ህፃናት ህይወት ማለፉ ተገጸ።	እሳት አደጋ (Fire Accident)
2	የገዳ ስርአት የአለም ወካይ ቅርስ ሆኖ በዩኔስኮ **ተመዘገበ**።	ተመዘገበ (Registered)
3	አትሌት ሀይሌ ገብረስላሴ የኢትዮጵያ አትሌቲክስ ፌዴሬሽን ፕሬዚዳንት ሆኖ **ተመረጠ**።	ተመረጠ (Elected)

As shown in Table 4.2, event triggers play a key role in event identification. Every word in news article can't be candidate event trigger, event trigger identification algorithm filter and use only possible word classes. Possible word classes are: verbs and nouns that describe events as they take place are considered typical features. We use part of speech tagger, morphological analysis and trigger list to select most informative triggers that can tell the occurrence of event.

Algorithm 4.4: Event Trigger Identification Algorithm

```
Input: Preprocessed news Article(A)
Output: event trigger
Begin
POST//Part-of-Speech Tagger
MA//Morphological Analysis
For every term in A
 If POS(term) in A == VB or NN
    Reduce term into its root form using MA
    Compare root word of candidate trigger with root of
    trigger list
    If trigger term match
```

```
        Compute candidate trigger position
        Compute TF(term)
        Compare the values
     The candidate trigger containing with highest
     value is recognized as key event trigger
    End if
  End if
End for
End
```

As indicated in *Algorithm 4.4*, we use trigger list and other local features indicated in algorithm to identify event triggers form news article. The true triggers head words are either verbs or nouns. Therefore, the algorithm considers only those words whose part of speech tags belong to these word classes as candidate triggers. After candidate triggers are identified the next task is identifying key event by computing the term of frequency value and position of each candidate trigger. Most of the time event extent located at beginning of news article, thus candidate located at first position is more important than the second candidate, and same way can be applied to other candidates depending on their position. Figure 4.3 shows event trigger identification workflow with example.

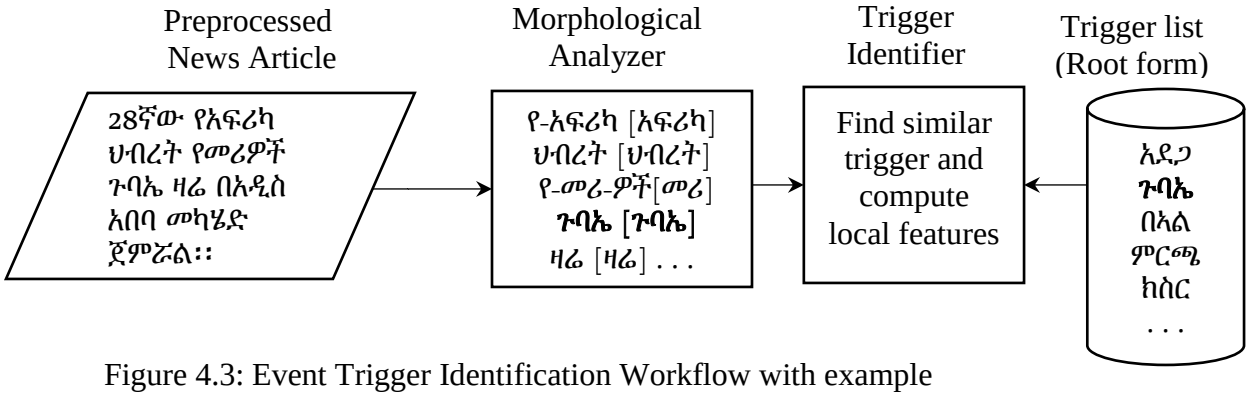

Figure 4.3: Event Trigger Identification Workflow with example

Morphological variability between candidate triggers extracted from news article and trigger list is handled using morphological analysis. Candidate trigger that have similar root word to the

trigger list root is labeled as event trigger. Specifically, when applied to candidate token *ct* and trigger list *Tl*, S(*ct, Tl*), checks whether *ct* is similar to any of the triggers in the trigger list. This is done using string similarity measure between a given *ct* and the trigger in trigger list *Tl.* For example, if a candidate trigger አደጋዎች is extracted from news article, first candidate trigger አደጋዎች is reduced to its root form using morphological analyzer resulting root word አደጋ. Then after string similarity between any of root words in trigger list is computed.

Other local features like term of frequency and position of candidate trigger has its own contribution to trigger identification. Term of frequency is the number of candidate trigger appear in news article. Candidate trigger with higher term frequency is more valuable than others. Position attribute is position where a candidate trigger first appears in the news article. A candidate trigger that occurs at the beginning of the news article is often more valuable than a candidate trigger that occurs at the end of that article.

4.8 Named Entity Recognition

Like part-of-speech tagger, named entity recognizer is a prerequisite for any information extraction process. Named entities are one of the most often extracted types of tokens during information extraction. After segmented tokens are tagged, the next task is recognizing the class of each entity identified in news article. To recognize different entities involved or mentioned in news article particularly which have relation with event mentioned, named entity recognizer is required. To extract event elements from news article we used manually tagged news named entities. The four event semantic elements are identified by using named entity recognizer. Event elements are categorized into different class of named entity like people, organization, place, time *etc*. Table 4.3 shows event elements and entity type along with their description.

Table 4.3: Event Elements, Possible Entity Classes and description

Event Element	**Entity type**	**Description**
Temporal elements	TIME	Describe time *when* (date, year, instant, etc.) the eventuality takes place
Spatial elements	LOC GPE	Describe place *where* an event happens

Participant(s) elements	PER ORG	Describe person or organization *who* participated during the course of event
Event extent (title)	--	Describe *what* is mentioned in news article

4.9 Event Elements Extraction

Event elements are arguments that tell entities involved in the course of event. Event elements extraction component is responsible to extract basic information that is event elements like extent, participant, temporal and spatial elements. Each element can be described in very different ways, and the values are captured by processing the news article. Typically, event elements are described in media using the facets of who, where, and when of news articles. Extracting event elements that tells the entity discussed in news article is more important in event extraction.

A number of entities can be mentioned in news article, but we only extract valuable elements those entities and values that occur within the scope of the event. To select event elements we use named entity recognizer and other local features like potential trigger and path from the extent to head word of the trigger. As shown in Table 4.2, event elements and their corresponding possible entity classes are presented. We have proposed different algorithms to extract event elements.

Our event element extarction algorithms first identifies all possible entities in a news article. Once the entities are identified, the distance of every entities toward the event extent is calculated. The one with the minimum distance from the event extent is considered the best candidate for particular event elements. Once we get the event elements candidate of an event, then we use rule based method to map these candidates to the 4W semantic elements of an event.

Event Extent Extraction

Event extent is a short textual description of an event. Event extent extraction corresponds to selecting sentence that best expresses the idea or mention of event. The extent extraction tries to pick the sentence which reflects the main content of the news article. Event extents are extracted using the result of measurement of the representativeness of the sentence. To determine the ranking of the sentences of a news article, a relative significance score is determined and the top ranked is chosen as event extent. Several features are used for ranking the sentences in the news article. We used a combination of various features of the sentence like position of the sentence in the news

article, length of the sentence, TF*IDF score of the sentence, number of named entities and presence of trigger words in the sentence. Algorithm 4.5 uses these features to rank candidate sentences.

The relative positioning of sentence, within a news article provides useful measurement of significance of the sentence. The existence of trigger words and number of named entities occurrence in news article furnishes a useful measurement of sentence significance. Each sentence is checked for the existence of trigger words. The existence of trigger words signals the mention of event and it adds to sentence score. Usually event elements (4W) are mentioned as named entities, as a result sentence containing greater number named entity get higher score. Also the long sentence holds more information than short sentences and scored higher. TF*IDF measures how frequently the words in a sentence occur relative to their occurrence in the entire news article.

Once each sentence is scored based on these features, the sentences are then ranked according to their weighting value. The highest score is assigned to the event extent. The score of a sentence can be calculated as a sum of various features of a sentence.

Algorithm 4.5: Event Extent Extraction Algorithm

```
Input: Preprocessed news article(A)

Output: Event extent

Begin

NER//Named Entity Recognizer

Trigger List

For every sentence in A

  Compute sentence position
  Compute sentence length
  Compute number of NEs
  Compute TF*IDF of sentence
  Find presence trigger words
  Compare weight sum
```

```
  sentence with highest value is considered as
  event extent
End for
End
```

Temporal Element Extraction

Temporal element extraction is responsible to extract date related expression from recognized named entities and select most relevant feature that relate to the occurrence of event. The temporal element is one of the key elements of an event which answers when event take place; its description can be a single day or it can be a time period with a starting and ending time. In order to determine the temporal references we use named entity recognizer and temporal normalizer.

Most importantly, there are two temporal coordinates of interest in event presentation, real world time and media time. The real world time refers to the absolute, unambiguous time when an event takes place in the physical world, while media time refers time when event published over media, that could be immediate, or after period of time. The relationship between real world time and the media time can vary widely across different news article. This is handled through capturing real time when event happened and media time separately, because the user may need of retrieving other media related to that event.

Algorithm 4.6: Temporal Element Extraction Algorithm

```
Input: preprocessed news article(A)
Output: event temporal element//time when event takes place
Begin
Named-Entity-Recognizer(NER)
Normalizer(Norm)
 For every term in A
   If NER(term) == TIME and RE matches //Time expression
```

```
      Compute distance from event trigger
       extract Event time
       extract Media time
      Select temporal element within or closer event extent
    End if
  End for
End
```

As described in Table 4.2, the class of temporal element is TIME. *Algorithm 4.6*, iterates over all sentences in news articles and collects the identified temporal elements. A number of date references can be extracted from a news article, but temporal elements are directly or the one closest to event extent is extracted. Some events don't contain any date references or not reliable, in such cases we decided to rely on the dates when the article was published. We could assume that an event is only being reported after it happens so in principle we can use the date of the article as the date of the event.

Spatial Element Extraction

Like temporal element, space or location is a key media variable to interpret events. As earlier discussed in Section 4.2 time and locations are the most important attributes for people to recall real world events. Similar to time, the description of location can also take many forms absolute, relative or approximate. Spatial element is absolute location where an event occurs. It answer question where an event takes place. Location can be a city, area or a whole country. Knowing location where event takes place is important task of event extraction process. In order to determine event location, we use named entity recognizer. As presented in Table 4.2 entities are categorized in specific classes, entity with class of location (LOC) and Geopolitical Entity (GPE) can be candidate for spatial reference.

Algorithm 4.7: Spatial Element Extraction Algorithm

```
Input: preprocessed news article(A)

Output: event Spatial element//place where event takes place

Begin

Named-Entity-Recognizer(NER)

 For every term in A

   If NER(term) == LOC or GPE

     Compute distance from event trigger

     extract location

     Select spatial element with in or closer to event extent

   End if

 End for

End
```

Spatial elements are extracted using *Algorithm 4.7.* The algorithm iterates over news article and search entity with class of LOC and GPE. Like temporal entity, a number of entities with these classes can be found, but element within event extent or closer gets higher probability.

Event Participant(s) Extraction

Event participants are entities that are involved during the course of an event. The specific types of participants that can be involved during the course of event are captured in this section. We capture participants, those entities which are mentioned within event extent and others. Most of the time event participant has close relation to event extent that is sentence from where event trigger is extracted.

People invest time, attention, and emotion while engaging in various activities in the real world event, for either purposes of awareness or participation. News services and social media platforms offer tremendous opportunities for people to become engaged in such real world events through information sharing and communicating about these events. These social media channels have

emerged as some of the most important platforms for people to report, share, and communicate with others about various types of real world events.

Most of the time users ask about whom it is discussing, one could ask, who participated in course of event? These questions are directly connected to the event itself (event capture and participation), on the other hand there is user engagement (interaction and influence) through communication or viewing. *Algorithm 4.8* show how to select event participants form preprocessed news article. Most common candidate of participant elements are person (PER), organization (ORG) or other classes of named entity recognizer.

Algorithm 4.8: Event Participant(s) Extraction Algorithm

```
Input: preprocessed news article(A)

Output: event participant element//entities involved

Begin

Named-Entity-Recognizer(NER)

 For every term in A

   If NER(term) == PER or ORG

   Compute distance from event trigger

   extract participant

  Select participant element with in or closer to event extent

   End if

 End for

End
```

To elaborate the above event elements extraction process in example, let as consider Amharic news article as an example.

“በአዲስ አበባ 18 ማዞሪያ ዛሬ ጠዋት የደረሰ ከፍተኛ የትራፊክ አደጋ የሶስት ሰዎች ህይወትን ሲቀጥፍ በርካቶችንም አቁስሏል። አደጋውን ያደረሰው ሲሚንቶ የጫነ ቱርቦ ከባድ ተሽከርካሪ ሲሆን፥ በአጠቃላይ በስድስት ሌሎች ተሽከርካሪዎች ላይ ጉዳት

ማድረሱም ተገልጿል፡፡ የተጠቀሰው ከባድ ተሽከርካሪ ከዊንጌት ወደ ቦሌ እየተጓዘ ነበር። ... አዲስ አበባ ፣ መጋቢት 26 ፣ 2008 (ኤፍ.ቢ.ሲ)"

From this article the event trigger identification and event elements extraction module extract the following information:

Trigger	ትራፊክ አደጋ
Extent	በአዲስ አበባ 18 ማዞሪያ ዛሬ ጠዋት የደረሰ ከፍተኛ የትራፊክ አደጋ የሶስት ሰዎች ህይወትን ሲቀጥፍ በርካቶችንም አቁስሏል።
Participant(s)	ሶስት ሰዎች
Location	አዲስ አበባ, 18 ማዞሪያ
Time	ዛሬ (መጋቢት 26 ፣ 2008)
Media time	መጋቢት 26 ፣ 2008
Publisher	ኤፍ.ቢ.ሲ

4.10 Classification

There are two types of event extraction approach, rule based and machine learning approach. Rule based require manually or automatically generated patterns or rules with pattern matching algorithm and it is suitable for domain dependent application. Rule writing requires enormous human effort and difficult to extract the rules automatically. Because of this reason we choose machine learning approach. Beside this, machine learning approach solve the problem of domain independence and consider event extraction as a classification problem, which focuses on the feature selection and classifier design. Thus, event extraction is typically classifier-based, and often uses training data set. The goal of classifier is to distinguish between event and non-event data, filter event related data sources from nonevent using deterministic feature values of input news article and training corpus.

In order to differentiate between triggers that represent events (positive instances) and those that do not (negative instances), or to choose the correct class label for every event candidate that represents an event. Machine learning algorithm integrated in our architecture makes use of various linguistic features. A feature corresponds to a specific property associated with an instance, and makes the connection between the instances that are observed and the categories that are to be predicted.

For this task, we used supervised machine learning algorithm. Supervised machine learning algorithm predicts the class of candidate event using features extracted from news article. In supervised machine learning, the labeling task can be defined as: given a set of n observations x_1, x_2, ..., x_n with their corresponding target class value y_1, y_2, ..., y_n, the goal is to predict the value of y for an unseen instance x. More formally, it can be defined as function f(x): x→y, where each instance x is represented as a vector of feature values, i.e., x = [f_1, f_2, ..., f_m], with *m* being the total number of features used in the representation. Depending on the number of distinct target values of y, one distinguishes between binary (with two target values) and multi-class classifications.

The class of candidate event is determined by the features of the event trigger and event elements. Therefore, event trigger and event elements (extent, temporal, spatial, and participating entities) are features to determine the class of candidate news article. The existence or associations of event trigger and event elements have their own contribution to determine the class of event. The machine learning algorithm is trained with these features, that tell the algorithm what is correct against incorrect extraction(supervision) through prepared training data.

To identify the class of candidate event into event and non-event, we use supervised machine learning algorithm called Maximum Entropy Classifier. Maximum Entropy is successfully applied in many natural language processing tasks, such as information extraction and syntactic analysis [69]. It is on the basis of maximum entropy classifier, it is used to model the known conditions and ignores unknown conditions. The classifier find probability distribution which satisfy all the known facts and it is not influenced by unknown facts[69]. Therefore useful features for the final classification can be freely increased without worring about their negative infuence. This merits force us to use the maximum entropy classifier.

Many features are considered when predicting whether a candidate belongs to a candidate class or not. Assume that X is an event vector containing features, and the label Y is true only if an event belongs to the candidate class. P(Y | X) denotes the probability of predicting a candidate event to be a candidate class, which is estimated based on the maximum entropy.

4.11 Event Representation

The main responsibility of this component is to map the extracted event facts to the designed event representation model. Traditionally events are represented in unstructured text, which is by the article in which they are expressed. As discussed in Section 2.5, a number of event representation

models are proposed by different researchers that can be used in various domains and different target applications. In this work, we choice ontology based event representation model, because ontology based event representation model provide semantic representation of event elements, implement deep concepts, properties and individuals, and relation in a formulated way.

As discussed in Section 2.6.3, the level of event representation has its own impact on human understanding or interpretation of events. In our work event structure is deeply covered for understanding the actual event information. Event representation is quite specific and includes not only actions, but also participants, time, and place.

Event definitions are described using different event elements. Only few of these elements are common for event definition like When (time) and Where (location). Minale A. et al. [42], considered three elements temporal, spatial and semantic to define and identify multimedia events. We adopt definition of the three event elements. In our work, we consider a four event elements the What (meaning) of the event, When (temporal), Where (spatial), and Who (participant). Event participant(s) are entities that are involved during the course of an event.

Ontology based event representation provide the basis for event representation that is semantically structured representation of events. Here we use ontology as a main mechanism to represent event information and construct event repository. On the basis of existing event model, we have designed event representation model for this work. The designed event representation model is capable to represent event semantic elements (4W) through defining concepts, properties and relations. We used Protégé to construct event representation model. Event centered modeling captures the dynamic aspects of an event. The visualization of proposed event representation model is presented in Figure 4.3, showing basic concepts and relations.

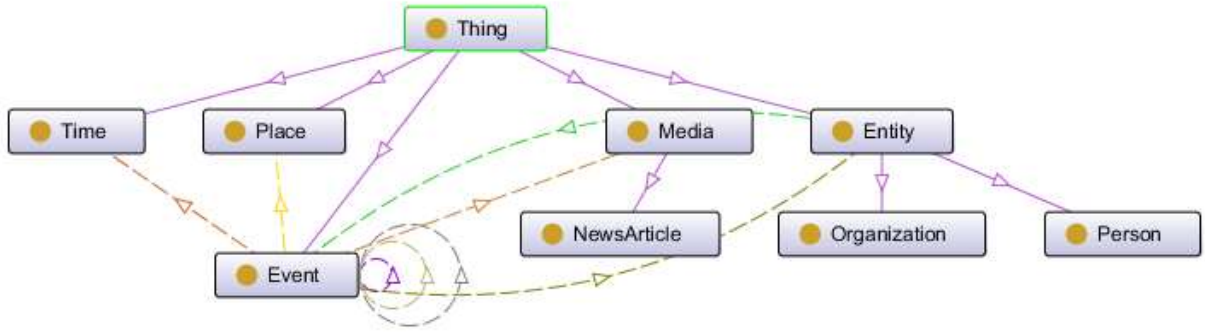

Figure 4.4: Visualization of Event Representation Ontology

4.11.1 Related Events Detection

There are multiple ways for an event to interact with other events, but it is impossible to list all relation between events. Event relation detection is challenging task due to the complexity of natural language. For this work we only consider generic relations like temporal, spatial, and informational aspects that can appear between events. Related event detection refers the task of identifying similar events. We can determine similar events using any of event trigger or properties like time, location, participant and triggers. Two events are related to each other, if they share same event trigger or event elements. If two events are triggered by same trigger and contain same event elements, then we assume that one of them is duplicate of the other.

Events are closely related to the notion of time and temporal relation may exist between two events. Since time commonality can be criteria to identify related events. Events take place in time, hence temporal relation between events are crucial to consider. Using a specific time we can identify very diverse events. Temporal relation between two events is analyzed by comparing time of occurrence. Location is related to place of occurrence of an event. Location is an important feature for identifying geographically related events, especially in combination with time. Spatial element is captured place where an event happened or occurred. It can identify events that touch various topics, which are relevant for a particular region.

Most of the described conditions offer a straightforward way for comparing event relations. Moreover, using combination of different properties, we can capture additional event relation such as spatial-temporal co-occurrence, temporal sequencing of events and *etc*. Using temporal elements of extracted event and event repository we can detect periodic event. By comparing the *'what'* and *'who'* of the events we can identify events that share similar participating entities or topics discussed in the events. Using such criteria we can identify similar events. Events structure necessarily contains the time period of the activity and its spatial characteristics. In addition to temporal and spatial characteristics, entities like people or objects that participate in an event are described.

The designed event representation model have ability to represent event through implementing concepts, properties and relationships presented in Section 5.4. Whenever new event is detected, similarity analysis is used to measure how much events are related to existing events. While performing similarity computation between events different event elements are considered.

Different measures use different sources of information to compute similarity scores. In our case we use previously extracted and stored event repository along with newly extracted event information. Event elements have significantly varying impact on the identification of similar events. We used the cosine similarity for textual features, $Sim_{extent}(E1, E2)$, $Sim_{location}(E1, E2)$, $Sim_{time}(E1, E2)$, and $Sim_{participant}(E1, E2)$. Each event contains four elements $<p_i, l_i, t_i, k_i>$, participants, location, time and keywords respectively. Pi can be defined as list of $<p_{i1}, p_{i2}, \ldots, p_{in})$ and same for other elements. Then the vectors of this event is considered for similarity measure with events $E_j <p_j, l_j, t_j, k_j>$ in event repository.

4.11.2 Mapping

Mapping is process of building event repository. It consists of adding new instances of concepts and relations into designed event repository. Event repository is a database of all recognized events. This process usually starts after the conceptual model of ontology is built. The mapper uses Word2vec predefined template to map the extracted event information into designed event ontology and finally store into event repository. The core idea of our event model is to take the extracted event facts from news article and provide a simple means of establishing a mapping between NLP and domain representation that is the concepts and relations.

Once event elements are extracted, then we used rule based methods to map these information into designed event representation model. The rules and mapping process is presented in Section 5.4. During the mapping process the extracted event elements are considered as instances of concepts in predefined conceptual event representation model. As a result, we can automatically generate an owl (owl/xml) file for extracted event facts, finally import them into event repository.

The resulting, event repository can then be used within any ontology enabled tool for further querying, reasoning, visualization, or other processing. All relevant event information are explicitly represented and stored, so that we can use event repository for higher level event semantic applications.

Chapter Five

Implementation and Experimental Results

5.1 Introduction

In previous Chapter, the design and implementation algorithms have been discussed. In this Chapter, the tools used for implementation that is tools for data preprocessing, event facts extraction, prototype development and ontology construction have been presented. In addition to this the evaluation and discussion of the proposed solution are also discussed.

5.2 Development Tools

Early form preprocessing task to representation we used different tools and programming environments. For text processing we use Python[23] and Natural Language Processing Toolkit (NLTK[24]) to preprocess news article, identify event trigger, and extract event elements from news article. NLTK is library for text processing and machine learning based on natural language processing. Python provides an excellent environment for performing basic text processing and feature extraction. Combination of Python and NLTK can easily add language aware data products to our larger analytical workflows and system.

In addition to preprocessing and feature extraction task NLTK has built in classification algorithms. We used NLTK machine learning classification algorithm called Maximum Entropy classifier. NLTK's Maximum Entropy classifier has object called *nltk.classify.maxent. MaxentClassifier*. The Maximum Entropy classifier in NLTK uses datasets to train the classifier and contain a train method which took input list as a feature, calculate the parameters for the probability distribution and finally return label.

For vector representation we used free python package called Gensim that contain an implementation of word2vec. The word2vec takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. The algorithms in gensim, such as Latent Semantic Analysis, Latent Dirichlet Allocation or Random Projections, discover semantic structure of documents, by

[23] https://www.python.org/

[24] www.nltk.org

examining word statistical co-occurrence patterns within a corpus of training documents. These algorithms are unsupervised, it only require a corpus of plain text documents. The resulting word vector file can be used as features in many natural language processing, machine learning and semantic representation applications.

To train word2vec model we used around 7000 words collected from various news articles in the FBC news datasets. 220 event triggering words collected from different source is also used as training data. For all unique event related words, the word representation vectors are collected from trained word2vec model. Each word is represented by an N-dimensional vector.

```
from gensim. models import Word2Vec
model = Word2Vec.load_word2vec_format(event, binary=True)
```

Other python package scikit-learn that contain t-SNE which is used for 2D or 3D visualization of vector. Using dimensionality reduction, a 2D map can be computed where semantically similar words are close to each other. Dissimilar high-dimensional vectors are distant in the two or three dimensional vector space. Here is sample code to include TSNE package.

```
from sklearn.manifold import TSNE
tsne = TSNE(n_components=2)
tsne.fit_transform(event)
```

Using dimensionality reduction, a 2D map can be computed where semantically similar words are close to each other. Once the statistical patterns are found, any plain text news article can be briefly expressed in the new, semantic representation, and queried for topical similarity against other news articles.

Protégé Beta version is used for ontology construction that is for the task of event representation. Our event representation model contain formally defined concepts, and properties that can represent event semantic elements. After event information is extracted, we can easily map the extracted event information into predefined event representation model. Python has *dicttoxml* package that map the extracted event facts form of python dictionary into xml format. In order to load the constructed owl ontology and integrated with other python module, we used *owlready* python ontology oriented programming package. Here is python and owlready code to load ontology on python editor.

onto_path.append("C:\Python34\event")

onto = get_ontology("http://www.event.com/ontologies/event.owl")

onto.load()

5.3 Data Collection and Preparation

To our knowledge, there are no suitable open Amharic news article dataset for open event extraction and evaluation. Thus, we collect Amharic news articles from Fana Broadcasting Corporate (Fanabc[25]) news archive starting from March 2014 to February 2017 using web crawler. Fanabc is a private news and information service located in Addis Ababa, Ethiopia. It provides international and Ethiopia related news in Amharic and other local languages on a daily basis. We choose Fanabc archive[26] because it is balanced news archive of different categories which focus on local and global issues.

To collect news article we used *HTTrack* which download the whole contents of the website. It contains a large number of articles and links to other webpages that were irrelevant to our purposes, thus we collect only the specific news articles that are useful for our purposes. The downloaded articles are originally formatted as individual html files within a page template for Fanabc website. We first stripped the entire html from the pages, including removing any template information such as the surrounding header, sidebar, and footer from the page. The resulting file represented only the article's content in plaintext. We first sampled around 17,000 news articles from the Fanabc, and discarded other news articles (e.g. non-Amharic, programs, advert, containing only an URL) to obtain a sample of 1225 event related news articles. Next, we manually identified event triggering (anchor) words that tells the mention of events, then trigger words are reviewed by expert. We identified total of 220 event trigger words and phrases. These event trigger words are used for event identification.

The task of data collection is followed by data preparation that is cleaning news article using techniques stated in Section 4.4.1. The preprocessing task includes tokenization, split up of each article into individual sentences and tokens. To do this, we used the SentenceTokenizer and WordTokenizer that comes with the NLTK package for sentence and word tokenizer respectively.

[25] www.fanabc.com

[26] http://www.fanabc.com/index.php/fbc-archive.html

The other task of preprocessing is normalization (character and temporal expression) as that is the format required of most statistical preprocessing. The task of character and temporal normalization is also implemented using python and NLTK packages.

5.4 Ontology Construction for Event Representation

In this section we demonstrate ontology development for event representation. Ontology development contains detail axioms which describe relation between classes, attributes and individuals are the main ontology development tasks. For this work we used Protégé 4.1 beta versions ontology editor. As discussed in Section 2.5 Protégé is most popular ontology editing and development tool. It provides GUI to create class, add object and data properties, and have visualization feature. In order to connect with other python modules we used *owlready* python ontology oriented programming package. It loads owl ontology as object, manipulate ontology classes, properties and instance, add python methods to ontology classes, and save to owl/xml format. To design event representation model we follow the basic ontology construction tasks.

A. Classes and Class Hierarchy Identification

Event representation ontology contains one super/top layer class called Thing. The middle layer includes: Event, Place, Time, Entity, Media, *etc*.in the bottom layer Person, Organization, NewsArticle, *etc*. middle layer classes inherits the super class (Thing). Figure 4.3, shows basic classes and class hierarchy of the event representation ontology. Table 5.1 shows the description of classes.

Table 5.1: Event representation ontology classes and description

#	Term Name	Type	Description
1	Event	Class	Is concept denotes dynamic eventuality. An event always has an event trigger which can be used to specify *what*. Property *hasId* used to uniquely identify event.
2	Place	Class	Describes spatial elements of an event. Properties *absLoc* or *relLoc* can be used to represent *where*.
3	Time	Class	Describes temporal elements of an event. Properties *eventTime and mediaTime* can be used to represent *when*.
4	Entity	Class	Is the superclass of all types of entities. It represent entities participated in event having several subclasses like person, organization, *etc.*
5	Person	Class	Is sub class of entity class, a concept to represent *who and whom*.
6	Organization	Class	Is a concept to represent *who and whom*.
7	Media	Class	Describes media aspect of an event
8	NewsArticle	Class	News Article is the media aspect of an event. Is subclass of Media. It has properties related to news article.

B. Object Property of Event Representation Ontology

Only classes can't answer many questions so we also need to define link inside or between these classes (such as properties) we use property which show relationship between individual to individual. We define object properties according to relationship between classes (as shown in Figure 4.3). Here are some of object properties, *involvedIn, inPlace, atTime,* and *etc.* for example object property *involvedIn* indicate relation between *Entity* and *Event* class, showing: entity *involvedIn* the course of event. Its domain is Entity and range is Event. If one attribute has many domain then, its domain is the intersection of its entire domain. The same pattern can be applied to other object properties. The sample description of event representation ontology object property is presented in Table 5.2.

Table 5.2: Sample Object Properties

#	**Term Name**	**Type**	**Description**
1	*involvedIn*	Object Property	Describe a (physical or social) object involved in an event, showing entity *involvedIn* the course of event, its domain is *Entity* and range is *Even*t.
2	*happenedIn*	Object Property	Describe place where an event happened, showing Event *happenedAt* Place, *Event* is domain and *Place* is range.
3	*happenedAt*	Object Property	Describe an abstract instant or interval of time when an event happened, showing Event *happenedAt* Time.
4	*mentionedIn*	Object Property	Describe a media or news article in which event is published, showing Event *mentionedIn* Media (news article).
5	*IsTempRelated*	Object Property	Holds temporally related events
6	*IsSpaRelated*	Object Property	Holds spatially related events
7	*IsSpaTempRelated*	Object Property	Holds spatial-temporally related events

C. Data Properties of Event Representation Ontology

In this step we identify data properties of event ontology which show the relationship between individual to data literal. Here are some examples of event ontology data property: *hasEventId, hasEventTrigger, hasEventExtent, hasPlaceName, hasTimeValue, hasPerName, hasOrgName, etc.* Table 5.3 shows sample data properties and corresponding domain and range.

Table 5.3: Sample Data Properties

Data Property	**Domain**	**Range**
hasEventId	Event	Literal String (uri)
hasEventTrigger	Event	Literal String
hasEventExtent	Event	Literal String
hasPlaName	Place	Literal String
hasTimValue	Time	DateTime
hasPerName	Entity	Literal String

D. Instances of Event Representation Ontology

Defining the instance (individual), first we have to load active ontology and select the right class, and then create its instances for the class. To identify right class for instance, we defined rule that assign the extracted event elements to specific class. Mapping task is driven by the rule keeping the defined relation (object and functional data properties) instance assignment. For this task we used *owlready* python package. Rules are defined to facilitate its mapping.

- Each class *Ci* is mapped into an abstract set that represents the set of all possible instances of the class.
- Each class *Ci* with individuals is mapped into an enumerated set containing these individuals.
- Each object property, relating two classes *Ci* and *C*j, is modeled as a constant. This constant is defined as a relation between the abstract sets C_i and C_j. The domain and the range of this relation are conform to the direction of the corresponding arrow.

As an ontology can be seen as defining the different data types of a domain, we suggest to translate it as a context component containing the following elements: it consists in deriving a specific types from domain ontologies to enrich the event repository. Here is sample *owlready* code to implement the rules.

```
onto = get_ontology("http://www.event.com/ontologies/event.owl")
onto.load()
EventId = "http://www.event.com/ontologies/event.owl#EV-00125"
```

```
    evid = Event(EventId)
    evid.has_extent.append(NewsArticle(get_event_extent()))
    evid.has_trigger.append(NewsArticle(Event_trigger_detector()))
    evid.happened_at_place.append(Place(Spatial_element_extraction()))
    evid.happened_at_time.append(Time(Temporal_element_extraction()))
    evid.has_participant.append(Entity(Participant_elements_extraction()))
    evid.published_on.append(Media(Event_source()))
    #Creating relation between functional property
    evid.mentioned_in = NewsArticle(Event_source())
    evid.has_place_name = Place(Spatial_element_extraction())
    evid.has_time_value = Time(Temporal_element_extraction())
    evid.has_participants = Entity(Participant_elements_extraction())
    evid.mentioned_in =NewsArticle(Event_source())
onto.save()
```

E. Reasoning Event Representation Ontology

To build correct and consistent ontology reasoning is most important part. Owlready HerniT reasoner is used to check consistency of event representation ontology and find the logic contradictions implicit in the definitions. The test of knowledge consistency includes detecting its reflexive, transmission and redundancy of knowledge.

The designed event representation model is shown in Figure 4.3, for the sake of clarity, only main concepts and properties are included. The general concepts of event model such as *Place, Time, Event, Entity, and Media* are used to represent an event and its 4W elements. Modeling media information is used as evidence to refer the detail information of events. Thus, by these we can connect the representation of the real world events in our model to the media assets that were taken during these events. We define concepts of *Media* and *NewArticle* to capture the characteristics of the news articles such as *URI, newsSource* and *publishedDate* corresponding to specific news article.

5.5 System Prototype

A prototype is developed to show the usability of the proposed work. The user interface allows the user to enter raw Amharic news article as shown in Figure 5.1. Once news article is

entered or uploaded the preprocessing module perform the task of tokenization and normalize news article expression into standard format.

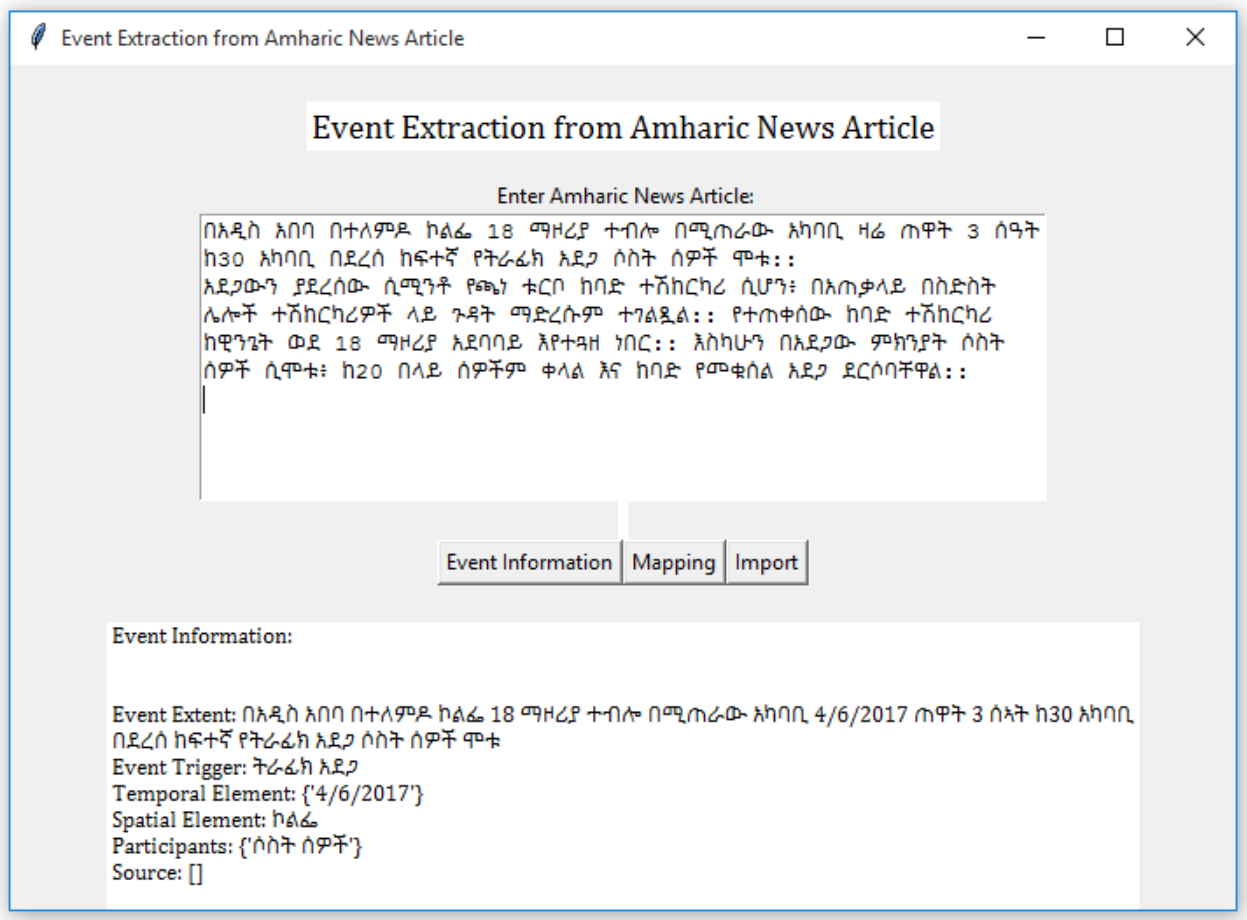

Figure 5.1: Prototype for Event Extraction from Amharic News Article

Following preprocessing module different operations like event information extraction, mapping the extracted facts into structured format and finally import the structured event facts into designed event representation model is done. As shown in figure 5.1, the system extract and display event information on user interface.

For event information extracted in Figure 5.1, the system map the extracted information to XML file format as shown in Figure 5.2. XML file is parsed by *dicttoxml* python package and imported into designed event representation model as instance.

Mapping Event Information into XML format:

```
<?xml version="1.0" ?>
<Event>
        <Event_Id type="str">EV-0011-6-2017</Event_Id>
        <inPlace type="str">ኮልፌ</inPlace>
        <Participants type="list">
                <item type="str">ሶስት ሰዎች</item>
        </Participants>
        <atTime type="list">
                <item type="str">11/6/2017</item>
        </atTime>
        <Source type="str">ኤፍ.ቢ.ሲ</Source>
        <EventExtent type="str">በአዲስ አበባ 18 ማዞሪያ 11/6/2017 ጠዋት የደረሰ ከፍተኛ የትራፊክ አደጋ የሶስት
ሰዎች ህይወትን ሲቀጥፍ በርካቶችንም አቁስሏል</EventExtent>
        <Event_Trigger type="str">ትራፊክ አደጋ</Event_Trigger>
</Event>
```

Figure 5.2: Result of mapping event information into XML format

5.6 Evaluation and Discussion

Our evaluation covers evaluating the performance of the implemented model through analysis of main components like key event identification, event element extraction, classification and representation. A manual tagging of news article has been made by seven annotators with the purpose of comparing it with automatically extracted by the system. Manual tagging is done by randomly selected four Computer Science postgraduate students and other three Information Science students. The task of manual tagging includes trigger identification, extent identification and event elements extraction. Manual tagging is carefully done by distributing up to 14 news article for a single annotator. For this reason, it is necessary to assure that the manual information is trustworthy and it does not alter the results of the evaluation.

To compute the performance measure of our system, we used most widely used information extraction measures like recall, precision and F-measure. Precision is the fraction of mentions predicted by the system that is correct, in that the mention is a valid mention of a real world event and linking to its participating event elements is correct. Thus the higher precision implies the better system performance. Precision is computed by using the following formula.

$$\text{Precision} = \frac{number\ of\ correctly\ extracted}{No.of\ correctly\ extracted\ + No.of\ wrongly\ extratced + No.of\ missed}$$

Recall is the fraction of mentions in the golden set of mentions that the system successfully extracted. Recall is computed by using the following formula.

$$\text{Recall} = \frac{number\ of\ correctly\ extracted}{number\ of\ correctly\ extracted\ + number\ of\ missed}$$

F-measure is the combination of precision and recall, it takes the interpretation of both. It is computed using the following formula.

$$\text{F} - \text{Measure} = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

5.6.1 Trigger Identification

To identify the mention of event in news article, we used guideline of ACE [60] and manually collected 220 event trigger words and phrases from various news domains. We evaluate the performance of trigger identification algorithm using manually tagged dataset. The evaluation is made with the parameters that compare the manually tagged and the one identified by the system. As previously discussed manual tagging include trigger identification, extent identification and event elements extraction. The basic idea her is, if the identified trigger contain the human tagged trigger, we mark the identification as correct, otherwise as wrong. We used precision as metric to evaluate the performance of our algorithm.

Evaluation result shows, the trigger identification obtain precision (67.1%). The experimental result also shows event triggers are more important to detect the mention of event and identification of event extent. Event identification use different features for event trigger identification, as a result trigger found at the beginning of news article is more important than trigger located at other position. The result also shows that our algorithm tends to be affected by the length of news article and number of candidate trigger identified in news article.

5.6.2 Event Elements Extraction

For event elements extraction evaluation, string similarity measure is used to compute the similarity between the extracted event information and the manually tagged result. We used 85 manually tagged news article by Addis Ababa University postgraduate students. Table 5.4 depicts evaluation result of event element extraction.

The result shows that, event element extraction obtain precision values for event extent extraction (68.2%), temporal element (77.6%), spatial element (70.6%) and participant (60.0%). Most often

news writer use common temporal expressions that is why we obtain higher precision value than other elements. In addition to this the experimental result shows that, the performance of event element extraction is directly influenced by the performance of named entity recognizer. Enhancing the named entity recognizer will improve the performance of event element extraction. The aggregate performance of our event element extractor obtain precision (69.1), recall (79.4%) and F-Measure (73.9%).

Table 5.4: Comparison of extracted event element with manually annotated articles

Event Elements	**No. of Similar Elements**	**No. of Dissimilar Elements**	**No. of Empty**	**Precision (%)**	**Recall (%)**	**FM (%)**
Event extent	58	5	22	68.2	80.6	73.9
Temporal Elements	66	5	14	77.6	91.7	84.1
Spatial Elements	60	12	13	70.6	83.3	76.4
Participant Elements	51	8	26	60.0	72.9	65.8
Average	235	30	75	69.1	79.4	73.9

5.6.3 Classification

We carried out evaluation of the classifier using dataset consisting of 1225 event related news article collected from Fanabc news archive. We reshuffled the collected news article, then we used the top 980 news articles as training data and the remaining 245 news articles are used to test the performance of the classifier. After training the Maximum Entropy classifier (*MaxentClassifier*) using training data, then we evaluate the performance of the trained classification algorithm using unseen test data. The classifier is evaluated using classifier built in function *nltk.classify. accuracy()* that calculate the accuracy of a classifier model with test data.

The accuracy measure is the percentage of inputs in the test data that the classifier correctly classified. The result shows, our machine learning classifier module correctly classifies (72.0%) of the events. The performance of classifier depends on the performance of feature extraction. Thus, with lower NLP tools our classier obtain good result. The result also shows that event trigger is more important than other elements, because it signals the mention of event.

To show importance of event elements in event identification, we consider different parameter values for weight parameters α, β, γ and δ, temporal (α), spatial (β), extent (γ) and participant (δ) elements when performing event definition and identification. Table 5.5 shows the FM results of by varying event elements weighting values.

Table 5.5: Importance of event elements in defining and extracting events by varying weighting values

Event elements weights	α (0.5) β(0.5) γ (0) δ (0)	α(0.33) β(0.33) γ(0.33) δ(0)	α(0.33) β(0.33) γ(0) δ(0.33)	α(0.3) β(0.3) γ(0.2) δ(0.2)	α(0.25) β(0.25) γ (0.25) δ(0.25)
FM	0.6063	0.6872	0.6853	0.7179	**0.7231**

Considering temporal and spatial elements only (neglecting extent and participant elements i.e. α = β = 0.5 and γ= δ = 0), we obtain lower value 0.6063. When neglecting extent or participant element, we obtain a better result than neglecting both elements. From the result we can understand that all four event elements seem to be equally important in identifying meaningful events, since the best result is obtained with a very close weight values for α, β, γ and δ.

5.6.4 Event Representation

Janez Brank *et al.* [73] classified ontology evaluation methods into four categories: comparing the ontology to a golden standard, using the ontology in an application and evaluating the results, comparisons with a source of data about the domain to be covered by the ontology, and evaluation by humans who try to assess how well the ontology meets a set of predefined criteria, standards, requirements, *etc.* We evaluate the representative ability of our event representation model using the third method that is comparing with source data about domain to be covered.

Our event representation model is evaluated with respect to the requirements, and event dimensions we considered in this work. The designed event representation model provides a basic vocabulary for semantic representation of event elements identified in news article. For this classes and properties are carefully selected to guarantee the model compactness as well as to supply rich semantics. It captures temporal, spatial, informational, experiential aspect of events and media information. Entities like people and objects that participate in an event are described. In addition to this, the designed event model is able to cover information of events in two levels, which are event information (action, time, place, entity and relation) and event media information. At same

time our model has more compact design with only few classes and properties, it is suitable for modeling Amharic news article.

Chapter Six

Conclusion and Future Works

6.1 Conclusion

Events are central aspect for human to perceive the real world eventuality, because it is a suitable unit in accordance with aspect of human perception and recognized as important metadata to fill the semantic gap between our experience of the world represented in media and its conceptualization. Event scenario is driven by the 4W's (what, when, where, and who) that is made up of event trigger and event elements. Traditionally events are published in news article or social media like mundane events and other ordinary news articles using language or other high level abstractions. Thus there is demand for automatic event extraction and representation model from unstructured free text.

In this study, we have proposed an event extraction and representation model from Amharic news article. Event modeling task start with processing unstructured news article. Different language specific processing has been done to transform the content of the news article into required and standard format. The task of event modeling involves five main task, preprocessing, trigger identification, event semantic elements extraction (4W), classification and event representation. Event triggers are identified by extensive linguistic analyze of text features and identified trigger list. To identify the mention of event, we used 220 event triggering words and phrases collected from various news domain. In order to extract event semantic elements, we use event triggering words, named entity recognizer and regular expression. A number of entities can be mentioned in news article, but we extract only entities related to key event that is closer to event trigger or found in event extent.

Once these tasks are accomplished, we used a machine learning classification algorithm to classify news articles telling about event into real world event and nonevent. We discover and classify events by exploring various features of extracted event trigger and event elements. Maximum Entropy classifier is used for this task. The classifier is trained by 1225 event related news article collected from Fanabc news archive. The identified events are semantically represented and stored in event knowledge base which can be used for various event level semantic applications.

For implementation we used different tools and development environments for preprocessing, event fact extraction, prototype development and representation. Different Python packages like

NLTK, owlready, *etc.* are used for text preprocessing, build classifier, develop prototype, ontology integration. Protégé beta version is used for ontology construction and visualization. We used owlready, python ontology oriented programming for mapping the extracted event information to constructed event representation model.

Evaluation of the proposed work is carried out by comparing manually tagged news article with the developed system. With lower NLP tools, our evaluation of event extraction model shows good performance. The event trigger identifier module obtain precision (67.1%) of event correctly which contributes to the better event element extraction and classification. The event elements extractor component shows greater obtaining precision (69.1%) while event classification module classify about (72%) of event correctly. In general, the proposed solution have shown a promising result when compared with highly resourced language event extraction systems.

6.2 Contribution of the Work

The main contributions of this research work are summarized as:

- The event mention identification from Amharic news article,
- Event element extraction, the usage of concept of 4W semantic elements in Amharic news article,
- Classification of published Amharic news article into events and nonevent,
- Event semantic elements representation model,
- Event modeling form Amharic news article.

6.3 Future Works

Event extraction from free unstructured natural language is a very complex task, which consumes more time, and can be enhanced by the performance of different NLP tools. Hence, there are a number of gaps for improvement and modification for event extraction and representation from Amharic text. Here are some of the recommendations we propose for future work.

- Incorporating Semantic Role Labeling: incorporating SRL, will bring additional relevant features that can be used to identify the role of participating entities. In addition to this additional event dimension like *why* and *how* can be extracted by reasoning.
- Enhancing NLP tools: we have used minimal NLP tools. The performance of NLP tools like POS tagger, named entity recognizer and morphological analysis have significant

impact on extraction process. Thus using enhanced NLP tools can improve the performance of the system.

- The relations between the events are very important for finding related event information. Currently we haven't yet put a lot of effort into identifying different kinds of possible relations between events. Identifying different types of relations between events like hierarchical events and events that form a storyline or periodic events.

References

[1] Wang Wei and Zhao Dongyan, "Chinese News Event 5W1H Semantic Elements Extraction for Event Ontology Population", *International World Wide Web Conference Committee (IW3C2), WWW 2012 Companion*, April 16–20, 2012, Lyon, France, 2012.

[2] Miller G., Beckwith R., Fellbaum C., Gross D. and Miller, K., "Introduction to WordNet: An online lexical database", *International Journal of Lexicography 3(4),* 235–312, 1990.

[3] Fazli Can, Seyit Kocberber, Ozgur Baglioglu Suleyman Kardas, H. Cagdas Ocalan and Erkan Uyar, "New Event Detection and Topic Tracking in Turkish", J*ournal of the American Society for Information Science and Technology,* October 1, 2009.

[4] Wei Wang and Dongyan Zhao, "Ontology-Based Event Modeling for Semantic Understanding of Chinese News Story", *NLPCC 2012*, CCIS 333, pp. 58–68, 2012.

[5] Farzind Aratefeh and Wael Khreich, "A Survey of Techniques for Event Detection in Twitter", *An International Journal of Computational Intelligence*, Volume 0, Number 0, 2013

[6] Nasser Alsaedi and Pete Burnap,"Feature Extraction and Analysis for Identifying Disruptive Events from Social Media", *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Cardiff School of Computer Science & Informatics Cardiff University, Cardiff, UK, 2015.

[7] George Valkanas, and Dimitrios Gunopulos, "How the Live Web Feels About Events", Dept. of Informatics & Telecom. University of Athens.

[8] Fausto Giunchiglia, Pierre Andrews, Gaia Trecarichi, and Ronald Chenu-Abente , "Media Aggregation via Events", University of Trento, Italy.

[9] Hidetsugu Nanba, Ryuta Saito, Aya Ishino, and Toshiyuki Takezawa, "Automatic Extraction of Event Information from Newspaper and Web Pages", *Springer International Publishing*, LNCS 8279, pp 171–175, Switzerland, 2013.

[10] Minale A. ABEBE, Fekade Getahun, Solomon Asresy and Richard Chbeiry, " Event Extraction for Collective Knowledge in Multimedia Digital EcoSystem", *IEEE AFRICON 2015*, At Addis Ababa, Ethiopia, September 2015.

[11] Zheng Chen and Heng Ji, **"**Language Specific Issue and Feature Exploration in Chinese Event Extraction", *365 Fifth Avenue*, New York, NY 10016, USA.

[12] Ansgar Scherp, Thomas Franz, Carsten Saatho and Steffen Staab, "A Core Ontology on Events for Representing Occurrences in the Real World", *Multimedia Tools and Applications manuscript,* Nov 18, 2010.

[13] http://www.omniglot.com/writing/amharic.html, Amharic Alphabet, Punctuation, and Numerals, last date accessed October 27 2016.

[14] Moges Ahmed and Sebsibe H/Mariam "Named Entity Recognition for Amharic Language", Unpublished Master's Thesis, Department of Computer Science, Addis Ababa University, Nov, 2010.

[15] Mesfin Abate and Yaregal Assabie (2014). "Development of Amharic Morphological Analyzer Using Memory-Based Learning", In *Proceedings of the 9th International Conference on Natural Language Processing (PolTAL2014), Springer Lecture Notes in Artificial Intelligence (LNAI), Vol. 8686, pp. 1-13,* Warsaw, Poland, 2014.

[16] Baye Yemam, *አጭርና ቀላል የአማርኛ ሰዋስዉ, የመጀመሪያ እትም,* 2002 ዓ.ም. ::

[17] Abeba Ibrahim and Yaregal Assabie (2013). "Hierarchical Amharic Base Phrase Chunking Using HMM With Error Pruning", In *Proceedings of the 6th Language and Technology Conference (LTC2013), pp. 328-332,* Poznan, Poland, 2013.

[18] Sisay Fissaha Adafre, "Part of Speech tagging for Amharic using Conditional Random Fields*", In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 47–54, Ann Arbor, June 2005.

[19] Getasew Tsedalu and Solomon Atnafu, "Information Extraction Model From Amharic News Texts", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, Nov, 2010.

[20] Michael Gasser, "HornMorpho: A System for Morphological Processing of Amharic, Oromo, and Tigrinya", Indiana University, Bloomington, Indiana, USA.

[21] Erkan Uyar, Fazl Can and Seyit Koçberber, "Near-Duplicate News Detection Using Named Entities", Unpublished Masters Thesis, Department of Computer Engineering, Bilkent University, May, 2009.

[22] Lars Asker, Atelach Alemu Argaw, Bjorn Gamback, Samuel Eyassu Asfeha and Lemma Nigussie Habte, "Classifying Amharic Webnews", Department of Information Science, Addis Ababa University, Addis Ababa, Ethiopia.

[23] Samir Elloumi, Ali Jaoua, Fethi Ferjani, Nasredine Semmar, Romaric Besanc xon, Jihad Al-Jaam, Helmi Hammami, "General learning approach for event extraction: Case of management change event", *Journal of Information Science* 39(2) 211–224. 2012.

[24] Raghu Anantharangachar1, Srinivasan Ramani, S Rajagopalan, "Ontology Guided Information Extraction from Unstructured Text", *International Journal of Web & Semantic Technology* (IJWesT) Vol.4, No.1, January 2013.

[25] Daya C. Wimalasuriya, Dejing Dou, "Using Multiple Ontologies in Information Extraction", *CIKM'09,* Hong Kong, China, November 2–6, 2009.

[26] Poonam Tanwar, V.Prasad and Kamlesh Datta, "Hybrid Technique for Effective Knowledge Representation and A Comparative Study", *National Institute of Technology*, Hamirpur, Himachal Pradesh, India.

[27] Aarti Singh and Poonam Anand, "State of Art in Ontology Development Tools", *International Journal of Advances in Computer Science and Technology*, Volume 2, No. 7, ISSN 2320 -2602, MMICT&BM, M.M.University Mullana, India, July 2013.

[28] Xiaomeng Su and Lars Ilebrekke, "A Comparative Study of Ontology Languages and Tools", Norwegian University of Science and Technology (NTNU), N-7491, Trondheim, Norway.

[29] Ansgar Scherp, Carsten Saathoff, Thomas Franz and Steffen Staab, "Designing Core Ontologies", *Applied Ontology 3 (2009) 1–3 1, IOS Press,* WeST, University of Koblenz Landau, Germany, 2009.

[30] Carl Lagoze and Jane Hunter, "The ABC Ontology and Model", Cornell University Ithaca, NY, October 24-26, 2001.

[31] Martin Doerr, Christian-Emil Ore and Stephen Stead, "The CIDOC Conceptual Reference Model – A New Standard for Knowledge Sharing ER2007 Tutorial*", 26th International Conference on Conceptual Modeling (ER 2007),* Auckland, New Zealand, 2007.

[32] Bhaskar Kapoor and Savita Sharma, "A Comparative Study Ontology Building Tools for Semantic Web Applications", I*nternational journal of Web & Semantic Technology (IJWesT),* Vol.1, Num.3, July 2010.

[33] Domingue, John, "Tadzebao and WebOnto: discussing, browsing, and editing ontologies on the Web", *In Eleventh Workshop on Knowledge Acquisition*, Modeling and Management, 18-23 April 1998.

[34] York Sure, Juergen Angele, and Steffen Staab,"OntoEdit: Multifaceted Inferencing for Ontology Engineering", Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany.

[35] Grigoris Antoniou and Frank van Harmelen,"Web Ontology Language: OWL", Department of AI, Vrije Universiteit Amsterdam.

[36] Asunción Gómez-Pérez and Oscar Corcho, "Ontology Languages for the Semantic Web", *IEEE intelligent systems*, 1094-7167/0, 2002.

[37] M. Andrea et al, "Formalizing Knowledge by Ontologies: OWL and KIF", CNR, IIT Department, Via Moruzzi 1, I-56124, Pisa, Italy.

[38] Jeff heflin, James Hendler, and Sean Luke, "SHOE: A Knowledge Representation Language for Internet Applications", Institute for Advanced Computer Studies, University of Maryland, College Park.

[39] Allan, J. Carbonell, G. Doddington, J.Yamron, and Y.Yang, "Topic Detection and Tracking Pilot Study Final Report", *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[40] Zhiwei Li, Bin Wang and Mingjing Li, Wei-Ying Ma, "A Probabilistic Model for Retrospective News Event Detection", *SIGIR'05*, August 15–19, 2005.

[41] Daniel Hienert and Francesco Luciano, "Extraction of Historical Events from Wikipedia", *ESWC 2012 Satellite Events*, LNCS 7540, pp. 16–28, 2015.

[42] Utz Westermann and Ramesh Jain, "E-A Generic Event Model for Event-Centric Multimedia Data Management in eChronicle Applications", Computer Science Department University of California Irvine CA 92697, USA.

[43] Nikolaos Gkalelis, Vasileios Mezaris, and Ioannis Kompatsiaris,"A joint content-event model for event-centric multimedia indexing", *Proc. Fourth IEEE International Conference on Semantic Computing (ICSC 2010)*, pp. 79-84., Pittsburgh, PA, USA, September 2010.

[44] Minale A. Abebe, Joe Tekli, Fekade Getahun, Gilbert Tekli, and Richard Chbeir, "A General Multimedia Representation Space Model Toward Event-based Collective Knowledge Management", *IEEE International Conference on Computational Science and Engineering*, 2016.

[45] R. Shaw, R. Troncy, and L. Hardman, "LODE: Linking Open Descriptions of Events", *4th Annual Asian Semantic Web Conference (ASWC'09)*, Shanghai, China, 2009.

[46] A.Scherp, T.Franz, C.Saathoff, and S.Staab, "F - a Model of Events based on the Foundational Ontology DOLCE+DnS ultralight", *International Conference on Knowledge Capturing (K-CAP)*, Redondo Beach, CA, USA, 2009.

[47] HeeMan Park, YoungLok Lee, BongNam Noh and HyungHyo Lee, "Ontology-Based Generic Event Model for Ubiquitous Environment", *International Journal of Innovative Computing, Information and Control*, Volume 5 ISSN 1349-4198, November 2009.

[48] Hage, W.R. van et al., "Design and use of the Simple Event Model (SEM)", *Journal of Web Semantics: Science, Services and Agents on the World Wide Web. 9*, 2, (2011), Netherlands, 2011.

[49] S. Sangeetha, R.S.Thakur, and Michael Arock,"Domain Independent Event Extraction System Using Text Meaning Representation Adopted for Semantic Web", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, ISSN: 2150-7988 Vol.2 (2010), pp.252-261, 2010.

[50] Hila Becker, Mor Naaman, and Luis Gravano, "Beyond Trending Topics: Real-World Event Identification on Twitter", *Association for the Advancement of Artificial Intelligence,* 2011.

[51] Hila Becker, Mor Naaman, and Luis Gravano, "Learning Similarity Metrics for Event Identification in Social Media", *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, August 25-28, 2015, Paris, France, 2015.

[52] David Ahn, "The stages of event extraction", *In Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, July 2006.

[53] Smriti Sharma, Rajesh Kumar, Pawan Bhadana and Sumita Gupta, "News Event Extraction Using 5W1H Approach & Its Analysis", *International Journal of Scientific & Engineering Research*, Volume 4, Issue 5, ISSN 2229-5518, May-2013.

[54] Shasha Liao and Ralph Grishman, "Acquiring Topic Features to Improve Event Extraction: in Pre-selected and Balanced Collections", *Proceedings of Recent Advances in Natural Language Processing*, pages 9-16, Hissar, Bulgaria, 12-14 September 2011.

[55] Hakan Kurt and H. Altay Guvenir, "Online Event Detection and Tracking in a Multisource Environments", M.S. in Computer Engineering, September, 2001.

[56] Ron Parka, James Allan and W. Bruce Croft, "Online New Event Detection, Clustering and Tracking", University of Massachusetts Amherst, PhD Dissertation, Department of Computer Science September 1999.

[57] Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji and Anette Frank, " Seed-Based Event Trigger Labeling: How far can event descriptions get us?", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 372–376, Beijing, China, July 26-31, 2015.

[58] Hristo Tanev, Jakub Piskorski, Maud Ehrmann and Vanni Zavarella , "Enhancing Event Descriptions through Twitter Mining", *Association for the Advancement of Artificial Intelligence*, 2012.

[59] S. Sangeetha, R.S.Thakur, and Michael Arock, "Domain Independent Event Extraction System Using Text Meaning Representation Adopted for Semantic Web", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, ISSN: 2150-7988, pp.252-261, 2010.

[60] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The Automatic Content Extraction (ACE) program-tasks, data, and evaluation", In LREC, 2004.

[61] Long Tian, W. Ma and Zhou Wen, "Automatic Event Trigger Word Extraction in Chinese Event", *Journal of Software Engineering and Applications*, 5, 208-212, Published Online December 2012.

[62] Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum, "Yago: A Core of Semantic Knowledge Unifying WordNet and Wikipedia", *In Proceedings of the 16th international conference on World Wide Web*, pp. 697–706 ACM, 2007.

[63] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, "Linguistic Regularities in Continuous Space Word Representations", *In Proceedings of NAACL HLT*, 2013.

[64] T. Mikolov, I. Sutskever, K.Chen, G.S. Corrado, and J. Dean., "Distributed Representation of Words and Phrases and their Compositionality", *In Advances in Neural Information Processing Systems*, pages 3111-3119, 2013.

[65] T. Mikolov, K.Chen, G. Corrado, and J.Dean, "Efficient Estimation of Word Representation in Vector Space", arXiv preparing arXiv:1301-3781, 2013.

[66] Hendrik Heuer, "Text Comparison using Word Vector Representations and Dimensionality Reduction", arXiv:1607.00534 v1 [cs.CL] 2 Jul 2016.

[67] Xiang-jun Wang, Swathi Mamadgi, Atit Thekdi, Aisling Kelliher and Hari Sundaram "Eventory -An Event Based Media Repository", Arts Media and Engineering Program, Arizona State University.

[68] Jakub Piskorski, Hristo Tanev, and Pinar Oezden Wennerberg, "Extracting Violent Events from On-Line News for Ontology Population", *BIS 2007*, LNCS 4439, pp. 287–300, 2007.

[69] Axel Magnuson, Vijay Dialan and Deepa Mallela,"Event Recommendation using Twitter Activity", RecSys'15, *ACM*, 978-1-4503-3692-5/15/09, September 16–20, 2015, Vienna, Austria, 2015.

[70] Hila Becker, Dan Iter, Mor Naaman and Luis Gravano, "Identifying Content for Planned Events Across Social Media Sites", WSDM'12, ACM 978-1-4503-0747-5/12/02, Seattle, Washington, USA, 2012.

[71] Konstantinos N. Vavliakis, Andreas L. Symeonidis, Pericles A. Mitkas, "Event Identification in Web Social Media through Named Entity Recognition and Topic Modeling", *In Science Direct, on journal of Data & Knowledge Engineering,* 2013.

[72] Qin Bing, Zhao Yanyan, Ding Xiao, Liu Ting and Zhai Guofu, "Event Type Recognition Based on Trigger Expansion", *Tsinghua Science And Technology*, ISSN 1007-0214 01/17 pp251-258, Vol 15, Number 3, June 2010.

[73] Janez Brank, Marko Grobelnik and Dunja Mladenić, "A Survey of Ontology Evaluation Techniques", In: *8th International Multi-Conference Information Society (IS 2005)*, pp. 166 170, 2005.

Appendix

Appendix I: Amharic Characters, Numerals and Punctuation Marks

A. Basic Amharic Scripts (ፊደል) [13]

	Orders									Orders						
	1st	2nd	3rd	4th	5th	6th	7th			1st	2nd	3rd	4th	5th	6th	7th
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ		h/ḵ	ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ		W	ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
h/ḥ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ		A	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ		Z	ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
s/ś	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ		zh/ž	ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
r	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ		Y	የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ		D	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
sh/š	ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ		j/ǧ	ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
k’/q	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ		G	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
b	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ		t’/ṭ	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
t	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ		ch	ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ
ch/č	ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ		p’/p̣	ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ
h/ḫ	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ		ts’/ṣ	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
n	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ		ts’/ś	ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
ny/ñ	ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ		F	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
a	አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ		P	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
k	ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ		V	ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ

B. Labialized Symbols [13]

ቈ	ቋ	ቌ	ኰ	ኳ	ኴ	ኋ	ኌ	ዀ	ዃ	ዄ	ዅ	ዏ
ጓ	ጔ	ጕ	ፙ	ፘ	ፚ	ᎀ	ᎁ	ᎂ	ᎃ	ᎄ	ᎅ	ᎆ
ᎇ	ᎈ	ᎉ	ᎊ	ᎋ	ᎌ	ᎍ	ᎎ	ᎏ	ᏸ	ᏹ	ᏺ	ᏻ

C. Amharic Numerals [17]

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

D. Amharic Punctuation Marks [13]

Punctuation Marks	Name	Description
:	Hulet Netib	Amharic word space
::	Arat Netib	Amharic full stop
፣	Netela Serez	Amharic comma
፤	Dereb Serez	Amharic semicolon
“ ”	Temiherte Tikes	Amharic quotation mark
!	Temiherte Anekero	Amharic exclamation mark
?	Timehrete Teyaqie	Amharic question mark

Appendix II: Sample Event Trigger List

ህይወት ኣለፈ
ሊግ ካፕ
ልደት በኣል
መስቀል በኣል
መሬት መንሸራተት ኣደጋ
ሙብረቅ ኣደጋ
መኪና ኣደጋ
መዉሊድ በኣል
መደበኛ ጉባኤ
ሙስና ወንጀል
ሙከራ ምርት
ማራቶን
ምርጫ
ሰላማዊ ሰልፍ
ሰሜን ለንደን ደርቢ
ሰርግ
ሳተላይት ማመንጠቅ
ሴካፋ
ስልጣን ለቀቀ
ስራ ጀመረ
ስራ ጉበኝት
ስርዓተ ቀብር
ስብሰባ
ሪዮ ኦሎምፒክ
ርእደ መሬት
ሪከርድ ሰበረ
ሹመት
ሽልማት
ሽብር ኣደጋ
ሽብር ጥቃት
ቀለበት ኣስራ
ቀብር ስነ ስርኣት
ቅርንጫፍ ከፈተ
በሞት ተለዩ
በርሊን ማራቶን
ቡና ደርቢ
ባሎንዶር ሽልማት
ባሎንዶር ኣሸናፊ
ባቡር ኣደጋ
ብር ሜዳሊያ
ቦምብ ጥቃት
ቦምብ ፍንዳታ
ቦስተን ማራቶን
ተመረቀ
ተመረጠ
ተመዘገበ
ታሰረ
ተሰረቀ
ተሸነፈ
ተሾመ
ተቃጠለ
ተከበረ
ተወለደ
ተዘጋ
ተፋታ
ታላቁ ሩጫ
ታመመ
ትራፍክ ኣደጋ
ቻምፒዮንስ ሊግ
ነህስ ሜዳልያ
ነጠላ ዘፈን
ንግስ
ኖቤል
ኣለም ዋንጫ
ኣሰልጣኝ መረጠ
ኣስመረቀ
ኣሸነፈ
ኣሸንዳ
ኣበረታች መድሀኒት
ኣተረፈ
ኣትሌቲክስ ሻምፒዮን
ኣንቁጣጣሽ
ኣዋጅ
ኣውሎ ንፋስ
ኣውሮፕላን ተከሰከሰ
ኣውሮፕላን ኣደጋ
ኣዉደ ራእይ
ኣዲስ በረራ
ኣእስ ምርት
ኣዲስ ምእራፍ
ኣዲስ ኣልበም
ኣዲስ ዝግጅት
ኣዲስ ግኝት
ኣዲስ ፈጠራ
ኣዲዋ በኣል
ኢጋባ
ኣጥፍቶ ጠፊ
ኣፍሪካ ዋንጫ
ኢሬቻ በኣል
ኢሮፓ ሊግ

ኤግዚቢሽን
እሳት አደጋ
እድገት ማእረግ
ኦሎምፒክ
ከሰረ
ከፍተኛ ዉጤት
ከፍታኛ ማእረግ
ኩፍኝ በሽታ
ክትባት
ኮንሰርት
ገና በኣል
ገዳ ስርአት
ጉባኤ
ጉበኝት
ጉዞ
ጉዳት
ወረርሽኝ
ወርቅ ሜዳልያ
ዋና አሰልጣኝ
ዋንጫ
ውድድር
ዘመን መለወጫ በኣል
ዚካ
የስራ ጉበኝት
የተመረዘ አልኮል
የተጠመደ ቦምብ
ግንቦት 20
ጎርፍ አደጋ
ጠቅላላ ጉባኤ
ጥምቀት
ጥቃት
ጦርነት አደጋ
የኔስኮ
የኔስኮ ቅርስ
ድርቅ አደጋ
ደርቢ
ዱባይ ማራቶን
ድያመንድ ሊግ
ጀልባ መስጠም አደጋ
ፕሮፌሰርነት ማእረግ
ጨዋታ
ፕሪሚየር ሊግ
ፋሲካ በኣል
ፍቹ ጨምበላላ
ፍንዳታ

Appendix III: Sample Code

Packages and Modules

```
import sys, os
import codecs  #used for utf-8 corpus reading /file
from nltk import *
from tkinter import *
import tkinter as tk
from math import *
import re
import l3
import textblob
from textblob import TextBlob
from textblob.classifiers import MaxEntClassifier
import numpy
from nltk.classify import MaxentClassifier
import dicttoxml
#import dict2xml
from xml.dom.minidom import parseString
from owlready import *
import logging
from gensim import corpora, models, similarities
from collections import defaultdict
import tempfile
import os.path
import datetime
import NER
import POST
import EventTriggerList
import Corpuses
import CharacterConvertor
```

Normalization

```
def characterNormalizer():
#ሀ, ሐ , ኀ, and ኸ ፣ ሰ and ሠ ፣ ፀ and ጸ ፣ ዐ and ዠ

    article = gettext.get(1.0,END)
    norm1 = re.sub('ሐ |ኀ|ኸ', 'ሀ', article)
    norm2 = re.sub('ሑ |ኁ |ኹ ', 'ሁ ', norm1)
    norm3 = re.sub('ሒ |ኂ|ኺ ', 'ሂ', norm2)
    norm4 = re.sub('ሓ |ኃ|ኻ', 'ሃ', norm3)
    norm5 = re.sub('ሔ |ኄ|ኼ ', 'ሄ', norm4)
    norm6 = re.sub('ሕ |ኅ|ኽ ', 'ህ', norm5)
    norm7 = re.sub('ሖ |ኆ |ኾ', 'ሆ', norm6)
```

```
    norms = re.sub('ሠ', 'ሰ', norm7)
    norms1 = re.sub('ሡ ', 'ሱ', norms)
    norms2 = re.sub('ሢ ', 'ሲ ', norms1)
    norms3 = re.sub('ሣ', 'ሳ', norms2)
    norms4 = re.sub('ሤ ', 'ሴ', norms3)
    norms5 = re.sub('ሥ ', 'ስ', norms4)
    norms6= re.sub('ሦ', 'ሶ', norms5)
# Normalizing the two Ts's
# ጸ ጹ ጺ ጻ ጼ ጽ ጾ:
# ፀ ፁ ፂ ፃ ፄ ፅ ፆ፧
    normTs = re.sub('ጸ', 'ፀ', norms6)
    normTs1 = re.sub('ጹ ', 'ፁ', normTs)
    normTs2 = re.sub('ጺ ', 'ፂ', normTs1)
    normTs3 = re.sub('ጻ', 'ፃ', normTs2)
    normTs4 = re.sub('ጼ', 'ፄ', normTs3)
    normTs5 = re.sub('ጽ', 'ፅ', normTs4)
    normTs6 = re.sub('ጾ', 'ፆ', normTs5)
# Normalizing the two A's
# ኣ ኤ ኢ  ኣ ኤ ኢ ኦ.
# ዐ ዑ  ዒ ዓ ዔ ዕ ዖ.
    normA = re.sub('ዐ', 'አ' , normTs6)
    normA1 = re.sub(' ዑ ', 'ኡ', normA)
    normA2 = re.sub('ዒ', 'ኢ ', normA1)
    normA3 = re.sub('ዓ', 'ኣ', normA2)
    normA4 = re.sub('ዔ', 'ኤ', normA3)
    normA5 = re.sub('ዕ', 'እ', normA4)
    normA6= re.sub('ዖ', 'ኦ', normA5)
    res.configure(text = "" +  str(normA6))
    return normA6
```

Trigger Identification

```
def Event_trigger_detector():
    TriggerList = EventTriggerList.TrueTrigger()
    newsnormarticle = get_event_extent()
    #normarticle = newsnormarticle.split()
    #for term in range(len(normarticle)):
    for trigger in range(len(TriggerList)):
        #pattern = re.compile(TriggerList[trigger], re.MULTILINE|re.U)
        matches = re.findall(TriggerList[trigger], newsnormarticle, re.UNICODE|re.MULTILINE)
        if matches:
            Tgger = TriggerList[trigger]
            return Tgger
```

Event Elements Extraction

```
def get_event_extent():
    TriggerList = EventTriggerList.TrueTrigger()
    newsSent = TemporalNormalizer().split('::')
    for trigger in range(len(TriggerList)):
        matches = re.findall(TriggerList[trigger], newsSent[0], re.UNICODE|re.MULTILINE)
        if matches:
            sent1_trigger = TriggerList[trigger]
            sent1_position = 1
            sent1_length = len(newsSent[0])
            sent1_ne = len(matches)
            return newsSent[0]
```

```
def get_event_extent_position():
    artSentence = TemporalNormalizer().split()
    cand_trigger = Event_extent_detector
    #normarticle = Preprocessor.TemporalNormalizer.split()
    for cand in range(len(artSentence)):
        if artSentence[cand] == cand_trigger:
            position = cand
    return position

def get_extent_length():
    artSentence = SentenceTokenizer().split()
    cand_trigger = Event_extent_detector
    #normarticle = Preprocessor.TemporalNormalizer.split()
    for cand in range(len(artSentence)):
        if artSentence[cand] == cand_trigger:
            length = cand
    return length
```

```
NER = NER.Named_entity()
count = 0
def number_number_of_ne():
    artSentence = SentenceTokenizer().split()
    for ne in range(len(artSentence)):
        artSentence(ne)
        count = count + 1
        return count

def Temporal_element_extraction():
    temporalElement = TemporalNormalizer()
    temporalElements = temporalElement.split()
    for term in range(len(temporalElements)):
        for key, value in NER.items():
            datepattern = re.findall(r"\d+/\d+/\d+", temporalElement, re.UNICODE|re.MULTILINE)
            matches = re.findall(key, temporalElement, re.UNICODE|re.MULTILINE)
            if datepattern:
                return set(datepattern)
            elif (key == temporalElements[term] or matches)  and value == 'TIME':
                event_time = key
                return event_time
```

```
def Spatial_element_extraction():
    spatialElement = TemporalNormalizer()
    spatialElements = TemporalNormalizer().split()

    for term in range(len(spatialElements)):
        for key, value in NER.items():
            matches = re.findall(key, spatialElement, re.UNICODE|re.MULTILINE)
            if (key == spatialElements[term] or matches) and (value == 'LOC' or value == 'GPE'):
                spatial_element = key
                return spatial_element

def Participant_elements_extraction():
    participantElement = TemporalNormalizer()
    participantElements = TemporalNormalizer().split()
    participant_elements= []
    for term in range(len(participantElements)):
        for key, value in NER.items():
            matches = re.findall(key, participantElement, re.UNICODE|re.MULTILINE)
            if (key == participantElements[term] or matches) and (value == 'PER' or value == 'ORG'):
                participant_elements = key
                return set(matches)
```

Event Representation

```
def Event_Information():
    EventExtent = get_event_extent()
    Eventtrigger = Event_trigger_detector()
    TempElement = Temporal_element_extraction()
    SpatialElement = Spatial_element_extraction()
    PartElement = Participant_elements_extraction()
    Source = Event_source()
    #label = Event_Classifier()
    event_info = ('\n\n''Event Extent: ' '{}' '\n' 'Event Trigger: ' '{}' '\n' "Temporal Element: "
    '{}' '\n' "Spatial Element: " '{}' '\n' "Participants: " '{}' '\n''Publisher: ' '{}'
    '\n' .format(EventExtent, Eventtrigger, TempElement, SpatialElement, PartElement, Source))
    if Eventtrigger != None:
        res.configure(text = "Event Information:\n" +  str(event_info))
        return event_info
    else:
        msg = 'Mention of event is not detected!'
        res.configure(text = "" +  str(msg))
        return msg
def Event_specification():
    event_specs = Event_Information()
    Event_trigger = Event_trigger_detector()
    res.configure(text = "Event Specification: \n" +  str(event_info))
    return event_specs
```

```
def dicttoxml_generator():
    EventExtent1 = get_event_extent()
    Eventtrigger1 = Event_trigger_detector()
    TempElement1 = Temporal_element_extraction()
    SpatialElement1 = Spatial_element_extraction()
    PartElement1 = Participant_elements_extraction()
    Source1 = Event_source()

    zare = datetime.datetime.now()
    EventId = 'EV-00'"%s-%s-%s" % (zare.day, zare.month, zare.year)
    eventdict = {
    'Event Trigger': Eventtrigger1,
    'Participants': PartElement1,
    'Source': Source1,
    'atTime': TempElement1,
    'EventExtent': EventExtent1,
    'inPlace': SpatialElement1,
    'Event_Id': EventId
    }
    xml = dicttoxml.dicttoxml(eventdict, custom_root='Event')
    dom = parseString(xml)
    dom1 = dom.toprettyxml()
    if Eventtrigger1 == None:
        msg = 'Mention of event is not detected!'
        res.configure(text = "" +  str(msg))
        return msg
    else:
        res.configure(text = "Mapping Event Information into XML format: \n\n" +  str(dom1))
        return dom1
```

Ontology Creation and Mapping

```
onto_path.append("C:\Python34\event")
onto = Ontology("http://www.event.com/ontologies/event.owl")
onto = get_ontology("http://www.event.com/ontologies/event.owl")
onto.load()
class Event(Thing):
    ontology = onto
ANNOTATIONS[Event].add_annotation(rdfs.comment,
"An event ontology that describes event and various event informations.")
class Entity(Thing):
    ontology = onto

class Person(Entity):
    ontology = onto

class Organization(Entity):
    ontology = onto

class Time(Thing):
    ontology = onto

class Place(Thing):
    ontology = onto

class Media(Thing):
    ontology = onto

class NewsArticle(Media):
    ontology = onto
```

```
class happened_at_time(Property):
    ontology = onto
    domain = [Event]
    range = [Time]

class happened_at_place(Property):
    ontology = onto
    domain = [Event]
    range = [Place]

class has_participant(Property):
    ontology = onto
    domain = [Event]
    range = [Entity]

class involved_in(Property):
    ontology = onto
    domain = [Entity]
    range = [Event]
    inverse_property = has_participant # inverse property

class has_trigger(Property):
    ontology = onto
    domain = [Event]
    range = [NewsArticle]
```

```
class has_extent(Property):
    ontology = onto
    domain = [Event]
    range = [NewsArticle]

class published_on(Property):
    ontology = onto
    domain = [Event]
    range = [Media]

class has_id(FunctionalProperty):
    ontology = onto
    domain = [Event]
    range = [str]

class has_Name(FunctionalProperty):
    ontology = onto
    domain = [Place]
    range = [str]

class has_Value(FunctionalProperty):
    ontology = onto
    domain = [Time]
    range = [str]

class mentioned_in(FunctionalProperty):
    ontology = onto
    domain = [NewsArticle]
    range = [str]
```

```
class hasName(FunctionalProperty):
    ontology = onto
    domain = [Person]
    range = [str]

class hasName(FunctionalProperty):
    ontology = onto
    domain = [Organization]
    range = [str]
```

```
#Creating relation using Property
def import_to_event_ontology():
    zare = datetime.datetime.now()
    EventId = 'EV-00'"%s-%s-%s" % (zare.second, zare.month, zare.year)

    evid = Event(EventId)
    print(evid)
    ctd = get_event_extent()
    ctd_event_extent = CharacterConvertor.CConvertor(ctd)
##    print(ctd_event_extent)
    evid.has_extent.append(NewsArticle(ctd_event_extent))

    etd = Event_trigger_detector()
    ctd_event_trigger = CharacterConvertor.CConvertor(etd)
##    print(ctd_event_trigger)
    evid.has_trigger.append(NewsArticle(ctd_event_trigger))

    see = Spatial_element_extraction()
    ctd_Spatial_element = CharacterConvertor.CConvertor(see)
##    print(ctd_Spatial_element)
    evid.happened_at_place.append(Place(ctd_Spatial_element))

    tee = Temporal_element_extraction()
    ctd_temp_element = CharacterConvertor.CConvertor(tee)
##    print(ctd_temp_element)
    evid.happened_at_time.append(Time(ctd_temp_element))

    pee = Participant_elements_extraction()
    ctd_part_element = CharacterConvertor.CConvertor(pee)
##    print(ctd_part_element)
    evid.has_participant.append(Entity(ctd_part_element))
```

```
    es = Event_source()
    ctd_event_source = CharacterConvertor.CConvertor(es)
##    print(ctd_event_source)
    evid.published_on.append(Media(ctd_event_source))

    #Creating relation between functional property
    evid.mentioned_in = NewsArticle(ctd_event_source)
    evid.has_place_name = Place(ctd_Spatial_element)
    evid.has_time_value = Time(ctd_temp_element)
    evid.has_participants = Entity(ctd_part_element)
    evid.mentioned_in =NewsArticle(ctd_event_source)
    onto.save()
    return 0
```

DECLARATION

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

Declared by:

Name: Bekele Abera Hordofa

Signature: _____

Date: _____

Confirmed by Advisor:

Name: Fekade Getahun (PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, February 2018.