2023-03

Entity Relation Extraction from Amharic Free Text using Deep Learning

Zelalem, Bekalu

Bahir Dar University

Bahir Dar Institute of Technology (BiT)

School of research and Postgraduate

Studies Faculty of Computing

MSc Thesis on:

Entity Relation Extraction from Amharic Free Text using Deep Learning

By:

Zelalem Bekalu

March 2023

Bahir Dar, Ethiopia

Bahir Dar University

Bahir Dar Institute of Technology (BiT)

School of research and Postgraduate Studies Faculty of Computing

Entity Relation Extraction from Amharic Free Text using Deep Learning

By:

Zelalem Bekalu Taye

A thesis is submitted to Faculty of Computing in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology in Faculty of Computing.

Advisor **Million Meshesha (PhD)**

March 2023

Bahir Dar, Ethiopia

Declaration

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation/ fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can evoke penal action from the sources, which have not properly cited or acknowledged.

Name of the student: **Zelalem Bekalu** Signature: __________ Date: __________

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

Advisor Name: **Million Meshesha (PhD)** Advisor’s Signature: million ____________________

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student **Zelalem Bekalu Taye** Signature______ Date 16.03.2023

As members of the board of examiners, we examined this thesis entitled "**Entity Relation Extraction from Amharic Free Text Using Deep Learning** "by ***Zelalem Bekalu Taye***. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science in "**Information Technology**".

Board of Examiners

Name of Advisor	Signature	Date
Million Meshesha (PhD)	million	16.03.2023
Name of External examiner	Signature	Date
Zewdie Mossie (PhD)		16.03.2023
Name of Internal Examiner	Signature	Date
Debas Senshaw (PhD)		16-03-2023
Name of Chairperson	Signature	Date
Mr., Abrham Debasu		04.04.2023
Name of Chair Holder	Signature	Date
Abdulkerim M (PhD)		16.03.2023
Name of Faculty Dean	Signature	Date
Mr., Asegahegn E		04-04-2023

Bahir Dar University Institute of Technology Faculty of Computing

Faculty Stamp

Acknowledgements

Primarily, I would like to thank God for his help and presence in my life and for giving the strength to do my final thesis.

Next, I would like to thank my advisor Dr Million Meshesha for helping me by giving deep comments starting from title selection. I would like to express my hearfelt gratitude to him, for his continual support. I greatly appreciate your valuable comments and support.

I would like to thank Dr., Seid Muhai Yimam Universität Hamburg House of Computing and Data Science (HCDS) And Department of Informatics Language Technology Group (LT) and my friends Haimanot Hailu, Bizuayehu Tadege, Mr., Mollalgn Yenealem, and to whom I did not mention for you for your assistance

Special thanks go to my family. You are always with me in my academic journey.

Finally, I would to thank my data annotators Ms., Genet Getachew and Mr., Getaye Asabu for giving in a timely manner and for effective work.

Table of Contents

List of tables

List of figures

List of algorithms

Abbreviations

Ad-MTL	Adversarial Multi-Task Learning
BiLSTM	Bidirectional Long Short Term Memory

CBOW	Continuous Bag of Words
CNN	Convolutional Neural Network
Conv1D	Convolutional layer
CRF	Conditional Random Field
DSR	Design Science Research
GLDR	Gated Linear Dilated Residual Network
KG	Knowledge Graph
MTL	Multi-Tasking Learning
NB	Naïve Bayes
NER	Named Entity Recognition
NLP	Natural Language Processing
POS	Part of Speech
QA	Question Answering
RE	Relationship Extraction
RNN	Recurrent Neural Network
SVM	Support Vector Machine
UD	Universal Dependency

Abstract

Relation extraction is a very useful task for several natural language processing applications, such as automatic summarization, knowledge graph development and question answering. An entity relation defined as a semantic interaction that holds between Named Entities. The entity relation extracting system developed for English or any other language in some specific domain cannot work for other languages of the same domain. Previously, there is research conducted on entity relation extraction from Amharic language free texts for a sentence have only one relation. When we study the Amharic sentence behaviors, it has two and more entities that exist so the relation between those named entities is also triple and multiple relations it has. This research attempted to design entity relation extraction from Amharic language free text with a sentence having multiple relations. Due to the number of relations between the named entities and the existence of triple and more relations of a sentence, to address the issues it is essential to develop a multi-label relation classification. Task-related entity indicators designed to enable a deep neural network to concentrate on the task-relevant information. By implanting entity indicators into a relation instance, the neural network is effective for encoding syntactic and semantic information about a relation instance.

We conduct the experiments using logistic regression classical machine learning algorithm in addition to encoder decoder models. These are LSTM, Bi-LSTM and CNN-BiLSTM. To conduct our experiments, we have used 2,500 sentences; it carries both triple and multiple relations. In order to extract our relation data, we have used entity indicators. To propose an optimal model with best relation classification from Amharic language entities unit we consider efficiency (training time, memory usage, and accuracy score). Finally, we have proposed multi-label relation classification using BiLSTM model with an accuracy score of 0.55. The major weakness of the study is unavailability of enough dataset to conduct an extensive experiment. As a result, there is a need to prepare corpora for conducting similar research.

Keywords: Named Entity, Relation Extraction, Entity Indicator Based Relation Extraction, and Deep learning approach, LSTM, BiLSTM and CNN-BiLSTM.

CHAPTER ONE ፥
INTRODUCATION

1.1 Background

An entity relation defined as a semantic interaction that holds between Named Entities (NEs) (Perera et al., 2020). Entity relationship usually involves two or more NE of a certain type (for example; Person, Organization, Location, Books, Time) and fall into a number of semantic categories (such as; የስራ_ቦታ, የተወለዱበት_ቦታ, መዳረሻ, ነዋሪ, መስራች, ጸሀፊው_ነው). Entity relation extraction begins with automating the system by locating individuals, locations, organizations, and entities in an unstructured text (Lv et al., 2021). Entity extraction, also known as named entity extraction, accomplished by combining rules described as entity lists, regular expressions, and NER algorithms for statistical modeling power. The ultimate goal of relationship extraction is to figure out what kind of association exists between two concepts in a sentence (Devisree & Raj, 2016)

This research mainly focuses on extracting the relation between named entities from Amharic language at the sentence level. The objective of the sentence X, which contains a pair of entities (E1, E2), so, the task is predicting the relationship R between those entities. E1 and E2 are entity, while R is a collection of predefined relations (D. Zhang & Wang, 2015).

Entity relationship extraction usually described as entity relationship triples <E1, E2, R> in which E1 and E2 refer to the entity type and R refers to the relation description text (C. Lv et al., 2021). After the preprocessing process of named entity recognition the relation triggers word recognition, the determined triples <E1, E2, R> are stored for further analysis or query. According to the definition, we divide the entity relation extracting tasks into three key parts, name entity recognition, relation word identification, and relation extraction.

Extracting entities and their semantic relationships from unstructured text is a big challenge for relation extraction. This large challenge is further broken into two well-known subtasks: named entity recognition (NER) and entity relation extraction (ER) (Zhong & Chen, 2021). Finally, entity relation extraction can define as follows. Given a sentence with annotated entities pairs e_1 and e_2, the task is to identify the semantic relation between e_1 and e_2 by a set of predefined relation classes

(For example, cause-effect, Entity-origin, component-whole, entity-destination, product-producer, member-collection, message-topic, content-container, instrument-agency and others).

Many scholars are interested in applying relationship extraction for discovering important information from large amounts of text in the age of big data. Entity relation extraction is a key task in information extraction that tries to extract a list of triplets from unstructured text that includes two entities and their semantic relationships (Peng & Chen, 2020).

Knowledge graphs (KGs, also known as knowledge bases or KBs) have recently become increasingly important in a variety of knowledge-driven applications, like, sentence generation (Trisedya et al., 2018), question answering (Dai et al., 2016), recommender systems (F. Zhang et al., 2016) and so on. Thus far, a number of large-scale KGs, such as Freebase (Bollacker et al., 1997), DBpedia (Auer et al., 2007), and YAGO (Tietz & Sack, 2019), have been built manually or automatically. However, the majority of the facts in them are abundant in the English version but scarce in other languages such as Amharic, implying that Amharic KGs less developed than English ones. As a result, research has been committed to obtaining structured information from unstructured texts using relation extraction in order to enrich Amharic KGs.

Entity extraction, relationship extraction, and event extraction are all types of data extraction (C. Lv et al., 2021). Entity extraction refers to approaches for detecting and identifying entities in text. Entity relationship extraction, on the other hand, detects the association between entities. Event extraction is the process of gathering knowledge about periodical incidents found in texts, automatically identifying information about what happened and when it happened. Relationship extraction, as the first stage in relation extraction, lays the technical groundwork for activities like knowledge graphs, intelligent information retrieval, and semantic analysis (C. Lv et al., 2021). As a result, relationship extraction approaches are useful not only for theoretical debate but also for practical implementation. Research on techniques to extract entities and their relationships can date back to the 1960s (C. Lv et al., 2021).

Traditional relation extraction has been proceeding by manual design and rule extraction. The traditional technique has two disadvantages (Qin et al., 2021). For starters, because the majority of entity pairs do not have relationships, there are numerous negative examples and an unequal relationship classification.

Secondly, overlapping triples are a serious problem. This is because appropriate training data cannot gather; learning becomes more complicated or perhaps impossible due to shared entities or many interactions between two entities. For instance, “Mr. Zelalem was born in Bahir Dar; a province in eastern Ethiopia” could be interpreted into “Mr. Zelalem was born in Bahir

Dar”, and “Bahir Dar lies in, eastern Ethiopia” the conventional algorithm cannot identify and classify properly without sufficient data (Qin et al., 2021)

Extraction of entity relationships provides fundamental support for knowledge graph, intelligent information retrieval, and semantic analysis promotes the construction of knowledge bases and improves the efficiency of searching and semantic analysis by identifying relationships among entities in natural language texts (A l A r f a j , 2 0 1 9) .

To extract associations between concepts in a text, researchers used ways based on co- occurrence statistics of specific phrases and machine learning approaches, as well as more linguistic approaches based on pattern or extraction rules, or hybrid approaches that combine these two techniques (AlArfaj, 2019).

Machine learning methods for extracting semantic relations can classified as supervised, semi supervised and unsupervised depending on the learning paradigm used (AlArfaj, 2019). The goal of supervised techniques is to figure out which types of relationships exist between ideas by employing predefined relationships. Support Vector Machine, Conditional Random Fields, and Maximum Entropy algorithms often learn to categorize new entity pairings into any of the relation types it has already observed.

Supervised learning approaches necessitate annotated training data and specified relationships. AlArfaj, (2019) proposed a semi-supervised method that uses labeled and unlabeled relation instances to learn semantic relations between named entities. Unsupervised learning approaches, in contrast to supervised, and involves inferring the patterns within datasets without reference to known or labeled, outcomes. Semi-supervised approach based on a small number of original seeds to obtain basic relations, a sample of language patterns or some target relation examples can employ until all of the target relations are discovered.

Relationship extraction can be done in a variety of ways, including text-based relationship extraction. These methods rely on the usage of pre-trained relationship structure knowledge or the learning of the structure to uncover linkages. Another approach to this problem involves the use of domain ontologies (Brambilla et al., 2006, Rindflesch et al., 2000). Visual detection of significant links in parametric values of items listed on a data table that shift locations while the table is permuted automatically as managed by the software user is another way. Structured resources such as semantic lexicons (e.g. WordNet, UMLS) and domain ontologies (such as the Gene Ontology) have inadequate coverage, rarity, and development costs, leading to new approaches based on vast, dynamic background knowledge on the Web.

According to Omar & Abdulla, (2021), identification of entities is a significant task that must complete correctly during the establishment of an ER documents written in Amharic text, and this work must completed. Entities, properties of entities and relationships must all extracted from natural language text in order to generate an ER document written in Amharic text. Such tasks support the steps for contributing knowledge outlined in full below.

- Creating a dataset, that machine learning classifiers can use to distinguish between nouns that represent entities and others.
- Extracting entities from natural language text using a machine learning technique.
- Developing a fully automated system that extracts entities relations from documents written in Amharic language text without no need of humans.

1.2 Motivation of the study

Amharic is serving as a working language of the Federal Democratic Republic of Ethiopia, Southern Nation Nationalities, and the Regional State of Amhara. Being an official working language, it used as a medium of instruction for primary and junior secondary schools. It is also a field of specialization at Diploma, Bachelor Degree, and Master's Degree levels at various universities in Ethiopia. Besides this, some literature works, newspapers, magazines, education resources, official credentials, and religious documents are published and available in the language. Hence, above all the alarming growth of information printed in Amharic language initiate to conduct this study**.**

As a matter fact, a lot of knowledge is available in unstructured Amharic text. News articles, messages, research paper may be machine accessible, but they cannot be used directly because the data in these texts is unstructured. However, it follows some rules, they may be semi- structured in web pages or structured in tabular form, but even natural language text follows grammar rules and some repeating patterns. The idea behind relationship extraction is that by exploiting these rules and pattern the data from these texts can extracted for further use. This is the other pushing factor to come across relationship extraction.

Further, in comparable with foreign languages, Amharic is one of the most resource scarce languages in context of NLP. Today the improvement in modern technology raises the availability of digital information on the Internet, which written by the Amharic language. Identifying relevant information from a given text manually is time consuming, error prone and tiresome task.

In general, no active research conducted on the automatic entity relationship extraction and a dramatic growth of electronic Amharic document from time to time are a motivating factor for this work to come up with solutions that can alleviate or minimize these problems.

1.3 Statement of the problem

A lot of valuable information produced in Ethiopia, most of them written in Amharic. The documents contain information related to research in many fields Worku (2015); particularly agriculture and water resource development; information on the development of the tourist and business sectors; government policies; and the bulk of information produced by offices in day-to-day work. Most government ministries, UN agencies, and NGOs also regularly produce informative bulletins, magazines, and newsletters. Information is available in abundance and a myriad of forms to an extent of making it nearly impossible to search manually, sift and choose relevant information. Therefore, valuable information must instead filter and extracted to avoid drowning in it. Triplet overlap is a complicated problem in an entity relation extraction, such as, **Zelalem** graduated from **Bahir Dar University**, and become a teacher there.

This sentence shows that, graduated school and workplace are the relation between Zelalem and Bahir Dar University. Different researchers proposed different methods that make relation extraction possible. Taghizadeh et al., (2018) proposed a cross-language method, which uses the training data of other languages and trains a model for relation extraction from Arabic text. The proposed method mainly relies on the Universal Dependency (UD) parsing and the similarity of UD trees in different languages. Regarding UD parse trees, all the features for training classifiers extracted and represented in a universal space.

Doshi (2018) presented a modified version of Deepdive for French language, which can be interesting for the application of non-English languages. Deepdive's Architecture consists of three phases, feature extraction, probabilistic engineering, and statistical inference and learning. Deepdive gets linguistic features by using tools like named-entity recognizer and dependency pathfinder. Then these features used to discover correlations between linguistic patterns and relations defined by the user. Such studies have their gap in extracting multiple relations in a sentence, co-occurrence between sentences.

The entity relation extracting system developed for English or any other language in some specific domain cannot work for other languages of the same domain. This is due to the reason that the relation extracting system has trained about the different nature of the language and the domain for which they are developed. Amharic is one of the widely used languages in Ethiopia, which has its own phonetics and grammar. In this regard, building an efficient relation extraction system for the Amharic language is an essential task. However, relationship extraction of Amharic texts falls behind the extraction of English, Chinese, French and other languages, because its complexity and difficulty. The complexity and difficulty come Redundancy of some characters: sometimes more than one character used for similar sound in Amharic (Worku, 2015). For example, the table below

Character	Other forms of character
አ	ዐ
ሰ	ሠ
ጸ	ፀ
ሀ	ሐ, ኀ

Table 1. 1The different forms of Amharic characters with homophone.

The problem of the same sound with various characters not only observed with core characters, but also exhibited in the same order of characters. For example, ሀ and ሃ, ኀ and ኃ; አ and ኣ; etc (Worku, 2015). The use of various forms of characters for the same sound poses a problem in the process of feature preparation for the classifier learning since the same word represented in different forms. For example, the word 'አገር' ('Country') represented in Amharic as አገር, ኣገር, ዓገር, ዐገር in addition the word 'ሀይማኖት' ('Religion') represented in Amharic as ሀይማኖት, ሃይማኖት,

ሐይማኖት, ሓይማኖት, ኀይማኖት, ኃይማኖት. Amharic characters with different forms of the same sound Character other form/s of the character.

One can imagine how the meaning of the original word diverted to different contexts. Spelling variation of the same word: the same word written in various forms (Worku, 2015). For example, the word ‘ሰምቶአል’ (‘he hears’) can be written in Amharic as ሰምቶአል, ሰምቷል, ሰምትዋል, etc. Spelling variation may happen also in the case of translating foreign word to Amharic. For instance, the word ‘ቴሌቪዥን’ (‘television’) written as ቴሌቭዢን, ቴሌቭዥን, ቴሌቪዥን, etc.

As sub task of information extraction Worku (2015) has introduced relation extraction for Amharic texts. In this work, the following gaps exist: Only the infrastructure domain supported, and it only retrieves relationships between named entities in the specified domain. For identifying those entities that lack a defined pattern, such as organization and location named entities, a gazetteer is used. As a result, the extraction is only valid for the entities listed or included in the gazetteer. Because different training datasets only cover a small portion of the available space, manual feature engineering takes a lot of effort and does not generalize well. It is therefore the aim of this study to develop Entity relationship extractor using machine-learning algorithms from documents written in Amharic language.

We review different published articles that attempt to extract relation extraction from text written in different languages however, none of them attempt to deal with triplet relationship extraction between the mentioned named entities. Our research mainly focuses on identifying triplet and multiple relationship that exist in a given sentence; for example, in a sentence, “ዘላለም በባህር ዳር ዩኒቨርሲቲ የመጀመሪያ ዲግሪውን ተመርቆ መምህር ሆነ ።” we have only two named entities, ዘላለም and ባህር ዳር ዩኒቨርሲቲ: the relationship between them is **graduated school** and **workplace.** For this kind of sentences, we prepare appropriate dataset for fixing triplet relationship. It is therefore the aim of this study to explore and design entity relationship extraction from free Amharic text. To this end, the study answers the following research questions.

- How should we identify the single relationship from triple relation and represent them for constructing a model using machine learning**?**
- Which machine learning algorithms are suitable for entity relationship extraction**?**

- To what extent the proposed prototype performs relation extraction from Amharic documents**.**

1.4 Objective of the study

1.4.1 General objective

The general objective of this study is entity relationship extraction from free Amharic text using deep learning.

1.4.2 Specific objectives

To achieve the general objective of this study, specific objectives given below targeted:

- To review related literature so as to identify suitable methods and algorithms
- To identify, collect and prepare a corpus of Amharic text
- To identify different representations of entity relationship in the Amharic language
- To develop a prototype using the selected optimal model
- To evaluate the performance of the proposed prototype

1.5 Methodology of the study

Methodology is a systematic process by which systematically solve the research problem Gondar & Universities (2019). This study aims to investigate and propose a relation extraction system between named entities in documents written in Amharic. A methodology is necessary to determine methods and approaches that should applied in the research process in a systematic and objective manner.

1.5.1 Research design

The methodology followed in this study is experimental research. Experimental design is the process of doing research in an objective and controlled manner in order to maximize precision and reach particular conclusions about a hypothesis statement. Because of this, a better conclusion made regarding the proposed hypothesis for extracting the relationship between named entities from documents written in Amharic. The following activities should apply systematically to achieve the study's goal:

1.5.2 Data collection and preparation

In this study, Amharic text corpus collected from Amharic news agencies, broadcasting media, online newspapers, Wikipedia, Blogs and magazines. These data have been organized and structured through cleaning, tokenization, and stop word removal in a way that they are suitable for experimentation. For experimentation, both the labeled and unlabeled Amharic text were prepared and used. Different facts about Amharic language like the grammatical structure and number representation conducted in order to understand the nature of the Amharic language with respect to relationship extraction.

1.5.3 Development tools

In order to develop an entities relation extraction model, different appropriate tools selected and used. We use Python programming language as a backbone of our experimentation, Tensor flow, amFlair for word embedding techniques in the conversion of from word to its vector form representing contextually for better understanding and extraction process.

a) Google colab Notebook

In this experimentation, we have used the google colab environment. Colab notebooks are Jupyter notebooks run in the cloud and integrated with Google Drive making them easy to set up, access, and share.

b) **Genism**

The Amharic text corpus trained using Genism for word vector generation, which is an open-source vector space modeling and topic-modeling toolkit implemented in Python. In Genism a corpus is simply an object, when iterated over, returns its documents represented as sparse vectors.

c) Tensorflow

Google's open-source machine learning library called Tensorflow. There are Python APIs in it. Although it has many abstraction capabilities, users may also be dealing with wrappers for computationally simple tasks like matrix operations, element wise math operators, and loop control. Tensorflow views networks as a directed graph of nodes that is wrapped with data flow computation and dependencies. Deep neural network classifiers developed using the Keras package, which uses Tensorflow as its back end.

d) Keras

Theano or TensorFlow can used as the back end for the Python deep learning machine-learning framework known as Keras. Its main objective is to make deep learning model implementation as quick and simple as feasible for research and development. For the proposed Amharic semantic RE system, deep neural network classifiers are created using Keras.

e) Pandas

Pandas is a Python package that presents an easy way of working with relational or labeled data by providing fast, flexible and expressive data structures. It has two primary data structures: Series (1-dimensional) and DataFrame (2-dimensional). The 2-dimensional data structure used in this study, which converted into 3-dimensional data, by using the NumPy package of Python. The LSTM and Bi-LSTM deep neural networks require a three- dimensional input; hence, the 2dimensional data changed into 3-dimensional data.

f) Python programming languages

Python is a powerful high-level, object-oriented programming general-purpose language. It has a wide range of applications from Web, scientific and mathematical computing to desktop graphical user Interfaces. In this experimentation process, we have used Python for deep learning for relation extraction tasks.

1.5.4 Testing and evaluation

The proposed system is tested and evaluated through the prototype using testing data (unseen data) to know how well it predicts the relation and to justify whether the proposed approach is outperforming the state-of-the-art system or not. In this study, we use F -score, precision and recall.

Precision: - is the number of true positive results divided by the number of all positive results, including those not identified correctly also known as positive predictive value.

Recall: - is the number of true positive results divided by the number of all samples that should have identified as positive also known as sensitivity.

F-score: - it measures the accuracy of the given model or system of the test.

1.6 Scope and limitation of the study

The aim of the study is an attempt towards relation extraction from Amharic free text. To this end, the scope of the study is detecting and extracting named entities, relation between entities and triple relation between them. Firstly, detecting the entities by named entity recognizer and encoding first entity out of pair then finding the corresponding entity and relation word; finally extracting the relation between them unless the system responds no relation between them. Secondly, if the sentence has more entity or not then if say find the second relation between those entities. Thirdly, the sentence has only two entities however; it may have either one or two relations. Other data types such as video, audio and graphic are not the focus of the study. The data set organized primarily from publications, news, blogs, and social media mainly focus on the history of a person with respect to place of birth, date of birth, graduation school, position, profession and so on. The method that we follow gathering the data is only concerned the above listed criteria. So many problems challenging the thesis not gone the planed one of them my computer hard disk crash down in addition my family health status.

1.7 Significance of the study

The study will facilitate biography development based on stories available in the form of text documents. It can also facilitate the database designers to easily understand and build the schema and the value of that schema. The main significant power of this study is to give short, precise information in a timely manner and limited space and used for further analysis.

Relation Extraction (RE) is of crucial significance to natural languages processing applications such as structured search, sentiment analysis, question answering, gene-disease, and summarization. For researchers working on NLP for further decision makings, such as in information extraction and text summarization tasks, this study is critical. This work used as a benchmark for conducting further research to successfully investigate and design a recommender system, as well as document summarization, because RE is a subtask of extracting information.

1.8 Thesis organization

This thesis is organized into five chapters The first chapter provides introduction to the study, with sub sections starting of background of study, motivation of the study, statement of the problem, objective of the study, methodology of the study, scope and limitation to the significance of the study. Chapter 2 presents a general understanding of relation extraction. This chapter introduces reviews made on different works of literature regarding relation extraction and the related subject areas together with its approaches. It discusses the Amharic language. Chapter 3 presents the Amharic text entity relation extraction model. This chapter also presents how the main components implemented. Chapter 4 presents the details of the experimentation and evaluation of the system. Chapter 5 concludes the thesis by outlining the contribution of the research and recommendation. It also shows some research directions that can be used in the future to improve the RE system for Amharic text.

CHAPTER TWO፡

LITRATURE REVIEW

In this chapter, the general overview of relation extraction presented. Different methodologies for relation extraction, key subtasks in relation extraction, assessment metrics, and an overview of Amharic language discussed in the sections below. In order to comprehend the scope of the work done, the section also covers natural language processing domains that connected to relation extraction.

The identification and categorization of semantic relationship mentions within a set of artifacts, often from documents written in Amharic Language, is required for a relationship extraction task. The RE is a sub task of information extraction. Various researchers have worked on the subject, but none of them has been able to fix the co-reference between entities and their complex relationship within a sentence, and research conducted on a language applied to another language, even if the domain is not applicable to another domain.

2.1 Information Extraction

Different IE systems for different languages and different domains using different approaches developed so far and are still on development but they all use the different task breakdown for IE. By the time that it ended in 1998 (Which is the end of MUC-7), the MUC program had arrived at a definition of IE split into five tasks.
These are:

- Named Entity Recognition
- Coreference Resolution
- Template Element Construction
- Template Relation Construction
- Scenario Template Production

2.1.1 Named Entity Recognition

The first step in most IE systems is the detection and classification of named entities, which are proper nouns, in a natural text. Named entity types to refer to places, persons, and organizations, and so on. Some applications may require the identification of other entity types, including products, proteins, genes, weapons, and others. It is all about finding entities.

Named entity recognition is the process of extracting nouns from a sentence or document that have a distinct meaning in the word itself, such as a person, organization, or location, and categorizing them accordingly (Sikdar & Gambäck, 2018). Traditional named entity recognition studies might separate into rule-based and dictionary-based investigations before utilizing machine learning. The majority of rule-based named entity recognition categorizes datasets using rules that manually defined by people. This strategy is mainly irregular and incomplete due to the nature of natural language, and it is extremely likely to function well only in particular datasets (Demeester et al., 2016; Lample et al., 2016). Dictionary-based named entity recognition classifies datasets using dictionaries that collected or dictionaries that created by the user. In some domains where less common language is frequently used, dictionary-based named entity identification is helpful for information extraction or retrieval. However, needing to manually organize dictionaries has its drawbacks, and because it is required to deal with continually changing and emerging new terms over time, managing dictionaries is costly, and processing non-pre-defined words has its restrictions.

Named Entity	Example
PERSON	ዘላለም, ብዙአየሁ, መንግስቱ
ORGANIZATION	ባህር ዳር ጨርቃጨርቅ, ባህር ዳር ዩኒቨርሲቲ
LOCATION	ባህር ዳር, አዲስ አበባ, ጅማ
DATE	25/02/2010, መስከረም 15
TIME	8:30 AM
PERCENTAGE	10%
MONETARY AMOUNT	$120.00, €250

Table 2. 1Named entity types as defined by MUC

Machine learning used in named entity recognition, as it is in many other fields of research. In recent years, there has been a lot of study on named entity recognition using deep learning. The Bi-LSTM-CRF model (Huang et al., 2015), which displays noteworthy performance in time series data utilizing supervised learning-based word embedding and non-supervised learning-based word embedding from a large corpus, is also used for named entity recognition.

A recent work on named entity recognition made extensive use of approaches based on pre learned language models from a huge corpus. Language models, as ELMo, Open AI GPT, and BERT are common. These models employ an accession mechanism, in which the encoder (SungMin et al., 2020) reuses sentences entered when the decoder predicts a word.

DQN is a reinforcement learning system that uses deep learning to improve Google's improved learning and utilized AlphaGo. Ref. presents a method for accomplishing the NER problem by employing policy-based active learning to apply reinforcement learning to CoNLL datasets. Ref. presents a strategy for applying the bidirectional LSTM-CRF model and DQN jointly to the Chinese NER challenge. This model performs well, especially when it comes to the news domain dataset (SungMin et al., 2020).

2.1.2 Coreference Resolution

Any given entity in a text referred to several times and every time it might referred differently. In order to identify all the ways used to name that entity throughout the document Coreference resolution performed. Coreference or anaphora resolution is the stage when for noun phrases it is determined if they refer to the same entity or not. There are several types of Coreference, but the most common types are pronominal and proper names Coreference, when a noun replaced by a pronoun in the first case and by another noun or a noun phrase in the second one.

Coreference resolution involves identifying relations between entities in texts. Besides entities identified by named entity recognition, this may also include anaphoric references to those entities. It is concerned with entities and references (such as pronouns) that refer to the same thing. Coreference resolution enables the association of descriptive information scattered across texts with the entities to which it refers.

2.1.3 Template Element Construction

Template element constructing task builds on named entity recognition and Coreference resolution. Its role is to associate descriptive information with the entities. It is all about what attributes entities have. The different recognized named entities will have different attributes for template element construction. Template element construction is domain dependent, as the types of information that are relevant depend on the types of entities that are important to the application domain. For example, relevant information about an organization includes whether it is private or public, if it is for profit or a charity.

2.1.4 Template Relation Construction

Before MUC-7, relations between entities were part of the scenario-specific template outputs of IE evaluations. In order to capture more widely useful relations, MUC-7 introduced the template relation task. The template relation task requires the identification of a small number of possible relations between the template elements identified in the template element task. This might be, for example, an employee relationship between a person and a company, a family relationship between two persons, or a subsidiary relationship between two companies. Extraction of relations among entities is a central feature of almost any information extraction task, although the possibilities in real-world extraction tasks are endless (Worku, 2015). It finds relations between template element entities. It is all about what relationships between entities there are.

The line between template entity and template relation is somewhat indistinct as both identify information relating to entities found by named entity recognition. What separates them is the domain of the application: Template relation needs relations between entities, with both the relation and entity types being relevant to the application domain. Template element needs additional information about entities, which may involve other entities but this data mainly used to enrich the description of the entity.

2.1.5 Scenario Template Production

Scenario templates are the prototypical outputs of IE systems, being the original task for which the term used. They tie together template element construction entity and template relation constructing relations into event descriptions. Scenario template is a difficult IE task; the best MUC systems score around 60%. The human score can be as low as around 80%, which illustrates the complexity involved (Worku, 2015). The scenario template-producing task is both domain dependent and, by definition, tied to the scenarios of interest to the users. Note however that the results of named entity, template relation and template element feed into scenario template.

2.2 Definition of Relation Extraction

With so much data available on the Internet, in offices, and in personal documents, it is critical to have certain technologies and tools to analyze it, derive information from it, and gain knowledge from it that may utilized for other reasons later. Thus, relation extraction is one of those technologies to obtain information from unstructured text extracting the association exists between two or more concepts in a sentence. The identification and categorization of semantic relationship mentioned within a set of entities, often from documents written in Amharic language, is required for a relationship ex traction task.

Finally, task of relation extraction can be divided into two phases (Igrejas et al., 2022): the task of detecting relations if a relation occurs between the corresponding entity mention and then after classifying the detected relation mentions into some predefined classes (cause-effect, Entityorigin, component-whole, entity-destination, product-producer, member-collection, messagetopic, content-container, instrument-agency and others).

Peng and Chen (2020) proposed end-to-end model based on a gated linear mechanism network and dynamic convolution to handle the challenge of entity–relation extraction. This model divided into two sections: E1 prediction and multi-turn E2 prediction. The encoder starts by converting the input sentence into a fixed-length vector, which it does using a 12-layer GLDR and dynamic convolutions.

In this phase, we will collect all of the sentence's E1s and place them in a "bag." Then, using a bidirectional LSTM layer, encrypt a sample E1 from the bag. This additional data is used to aid in the prediction of E2s and their relationships. In particular, there is a prediction for the position of E2 for each predefined relationship. In other words, we can anticipate both E2s and relations at the same time, as well as deal with situations when relations overlap. The paper conducts experiment on two widely used datasets called NYT and WebNLG (Tan et al., 2017). The experiments measure by precision, recall and F-measure. The experimental results reveal that our strategy outperforms baseline methods, demonstrating that it is successful.

2.3 Features of Entity Relation Extraction

Liu et al., (2021) relationship extraction differs from other NLP tasks such as sentiment analysis and news classification in three ways.

Firstly, Entity Relationship Extraction is applicable to a wide range of disciplines. Typically, researchers concentrate on a single domain or a small number of domains. Traditional techniques rely on rules, dictionaries, and ontologies to solve problems with limited relationship categories. Supervised, semi-supervised, and unsupervised models are examples of machine learning-based methodologies. Recently, supervised and remote supervised models added to the list of deep learning-based techniques. All of these models are simple to construct, but they lack portability and extensibility.

Secondly, Entity Relation Extraction requires heterogeneous data. Data can be structured, semi structured, or unstructured, and it can come from a variety of sources. Deep learning typically used with structured data; non-supervised aggregation methods typically used with nonstructured textual data due to the unpredictable nature of relationship categories; and semisupervised or distant supervised methods typically used with semi-structured data like Wikipedia.

Lastly, Entity Relation Extraction must deal with a variety of relationships, which might result in data noise. Although there are many different types of links between entities, early research frequently overlooked them and failed to account for hidden relationships. In recent years, the use of graph structures in relationship extraction has brought in a new technique for dealing with entity and relationship overlaps. It discovered that utilizing a small number of adversarial instances can avoid model overfitting, and proposed to utilize adversarial training to improve model performance to deal with data noise.

2.4 Approaches of entity Relation Extraction

There are two main approaches to the design of Entity Relation Extraction systems, such as knowledge engineering approach and Machine learning approach (Worku, 2015).

2.4.1 Knowledge engineering approach

Grammars expressing rules for the system developed by hand utilizing knowledge of the application domain in the knowledge engineering technique. A person who designs such a system or is in charge of defining the rules (i.e., a knowledge engineer) must be an expert in the knowledge domain extracted, or at the very least have a good understanding of it.

The knowledge engineering strategy, in addition to demanding talent and thorough knowledge of a specific RE system usually necessitates a lot of labor, a long test-and-debug cycle, and it is reliant on having linguistic resources on hand, such as adequate lexicons. Building the rules via a knowledge engineering technique takes a long time, and the system difficult to maintain. The majority of the best-performing systems, on the other hand, are hand crafted. The computer system does not learn anything from the data in this technique.

It solely uses what human experts have discovered. According to Appelt and Israel (1999), a knowledge engineering approach is a critical component in developing a high-performance system. Knowledge engineers will build every area of knowledge, resulting in high-level performance. Using this method, creating a system is an iterative process. To begin, the knowledge engineer creates a specific rule. Then he tests it against the available texts to see if it works appropriately. If necessary, changes made, and the rule re-evaluated until a satisfactory outcome obtained. In some instances, it is termed as rule-based approach, since it involves writing rules.

2.4.2 Machine Learning Approach

There is no need manually construct extraction rules when using a machine learning approach. As a result, a person in charge of the relation extraction process does not need to understand how to develop rules or how a system operates. Those rules created using a machine-learning algorithm applied in the relation extraction system. To accomplish so, the algorithm needs to have access to a large number of training texts in the subject of interest. Because machine learning learns and works based on training data, a huge corpus utilized to train the system for greater performance. This method also known as the automated training method.

Rather than concentrating on the creation of rules, the automatic training method concentrates on the training data. While developing a system using a machine learning approach is faster than using a knowledge engineering approach, it does necessitate a large amount of training data. As long as corpora of domain-related texts are accessible, the same machine-learning algorithm can be used to different domains in this method. As a result, unlike the knowledge engineering approach, machine learning is domain agnostic.

It is feasible to come up with criteria that determine which strategy to choose based on an analysis of the benefits and downsides of both approaches (Hoos, 2020). The presence of a set of relevant texts that utilized to train the algorithm is the most crucial prerequisite for choosing the automatic training strategy. The availability of a person with experience designing extraction rules is the most important factor in the knowledge engineering approach. There are three types of automatic learning systems: supervised learning systems, semi-supervised learning systems, and unsupervised learning systems.

A. Supervised methods

Methods that supervised and based on a training set with domain-specific examples that tagged. This based on a completely labeled corpus. Relation extraction treated as a classification task in this approach. Support Vector Machines (SVM), Conditional Random Fields (CRF), decision tree, and maximum Entropy are some of the most commonly utilized supervised algorithms (MaxEnt) (Mahendran, 2022).

A recent attempt was made to extract the relationships between Arabic NEs (Alotayq, 2013) who made use of a MaxEnt-based classifier. When applied to the ACE corpus, this method delivers satisfactory results based solely on morphologic and part of speech (POS) information. The main disadvantage of these methods is that creating a properly annotated corpus can take a long time and effort. On the other hand, if training data is available, readily apply these systems to a different area. The supervised machine-learning model must determine whether E1 and E2 have any relationship (R). As a result, the task of relation extraction becomes the challenge of relation detection in a supervised technique. In summary, supervised RE is accurate but largely required on manually labelled data.

B. Unsupervised methods

Unsupervised Learning systems lighten the user's load by requiring simply a declaration of the required data. The user does not provide any extraction patterns in advance. Annotated corpora not employed in this learning to increase the system's performance. The most difficult task is to translate the user's requirement into a set of extraction patterns. The systems work by growing a small set of extraction patterns using bootstrapping approaches.

C. Semi-supervised methods

In entity RE systems, another method called semi-supervised learning used to deal with the still high need on human expertise in supervised learning. A system learns from a mixture of smalllabeled (annotated) data and sufficient unlabeled data using semi-supervised learning (Hoos, 2020). A small-labeled data set coexists with a large unlabeled data set in many applications. It is not a good idea to train the system with only a tiny-labeled data set because it is widely known that when the ratio of training samples to feature measurements is small, the training result is inaccurate.

To boost performance, the system must blend labeled and unlabeled data during training. The unlabeled data utilized for density estimates or labeled data preprocessing, such as determining underlying domain structure. In other words, the system extracts patterns from annotated data and uses those patterns automatically classify unannotated data. As a result, all data for the training labeled. Semi-supervised learning saves time and effort while delivering results that are comparable to supervised learning.

D. Reinforcement learning

Reinforcement learning is a form of machine learning method in which a smart agent (computer program) interacts with its surroundings and learns how to operate in that environment (Naeem et al., 2020). Reinforcement Learning is a feedback-based Machine Learning technique in which an agent learns how to behave in a given environment by executing actions and seeing the outcomes of those actions. The agent receives positive feedback for each excellent action, and negative feedback or a penalty for each bad action in which case the agent expected to learn again. Unlike supervised learning, the agent learns autonomously utilizing feedback and no labeled data in Reinforcement Learning.

2.5 Deep learning approach

NLP tools are required for feature extraction in the conventional, non-deep learning techniques to relation extraction. The performance of relational extraction may influence by the faults made by those NLP tools, which can magnify in relational extraction. Deep learning techniques can reduce these inaccuracies. Deep learning often referred to as deep structured learning or hierarchical learning is a development of machine learning techniques that aims to create a layered model of inputs more usually referred to as neural nets (Alom et al., 2019; Igrejas et al., 2022).

Deep learning algorithms advantageous for better understanding from the complicated structures from large dataset in backpropagation process to adjust their internal parameters (Aschenaki Abi Abera, Yaregal Assabe, Mesfin Kifle, 2020). This unique feature of deep learning enables the model to extract relations more accurately. Numerous scholars have recently used relational extraction to apply depth-learning techniques. End-to-end models, dependency models, and remotely supervised models are the three categories used to categorize the current trend in deep learning models for relation extraction. Among them, we briefly discuss only the models used in this thesis.

End-To-End Models

Instead of breaking the problem down and attempting to solve smaller difficulties, end-to-end models are effective means of learning to handle the challenge at hand (Miwa & Bansal, 2016).

Traditional methods for relation extraction, for instance, frequently rely on a pipeline of the two discrete subtasks of entity recognition and relation extraction. It first finds mentions of named entities before performing relation extraction on those mentions. Entity recognition's result used as relation classification is input. End-to-end relation extraction refers to the two subtasks taken together without taking into account their underlying interdependencies. It recently been suggested to use end-to-end models that lack high-level characteristics to prevent error propagation from entity recognition to relation extraction. Convolutional neural networks (CNN) and recurrent neural network (RNN)-based End-to-End deep learning models frequently employed for relation extraction. CNN is not appropriate for learning distant semantic data
(Grishman, 2015). Thus, our approach is RNN based (Burget, 2010) specifically Attention-based Bi-directional LSTM model.

Recurrent Neural Networks (RNN) based methods

The consideration of the sequential relationship between inputs and outputs is a significant constraint of traditional neural networks. Every input and output thought to be independent of one another. Recurrent neural networks (RNNs), which have demonstrated remarkable performance in many NLP applications, are proposed to get around this constraint (Burget, 2010). Recurrent neural networks employ their recollection of previous calculations to inform their current output computation. An example of a recurrent neural network.

Long Short-Term Memory (LSTM)

LSTM has achieved the best-known results in relation extraction (D. Zhang & Wang, 2015). Hochreiter and Schmidhuber (Cascade-correlation & Chunking, 1997) to overcome the gradient vanishing problem firstly propose LSTM units. A unique class of RNN model called Long Short-Term Memory (LSTM) created to address backflow issues (Cascade correlation & Chunking, 1997). The LSTM model, which Graves has recently enhanced and advocated, can solve the long-distance reliance issue with RNN (Nandanwar, 2021) In addition, CNN.

Memory blocks are a group of recurrently connected blocks that make up the LSTM layer. Each one has one or more memory cells with recurrent connections. They specifically made to address the issue of long-term dependencies when more background knowledge is required for the current activity. The cell state, which meticulously controlled by structures known as gates, can be altered by the LSTM by removing or adding information. Gates regulate the information flow. LSTM units typically implemented in blocks of several units. The three gates in these blocks— input, forget, and output—provide continuous analogs of write, read, and reset operations for the cells and regulate information flow by using the logistic function (Rengasamy et al., 2020; Rybalkin et al., 2021).

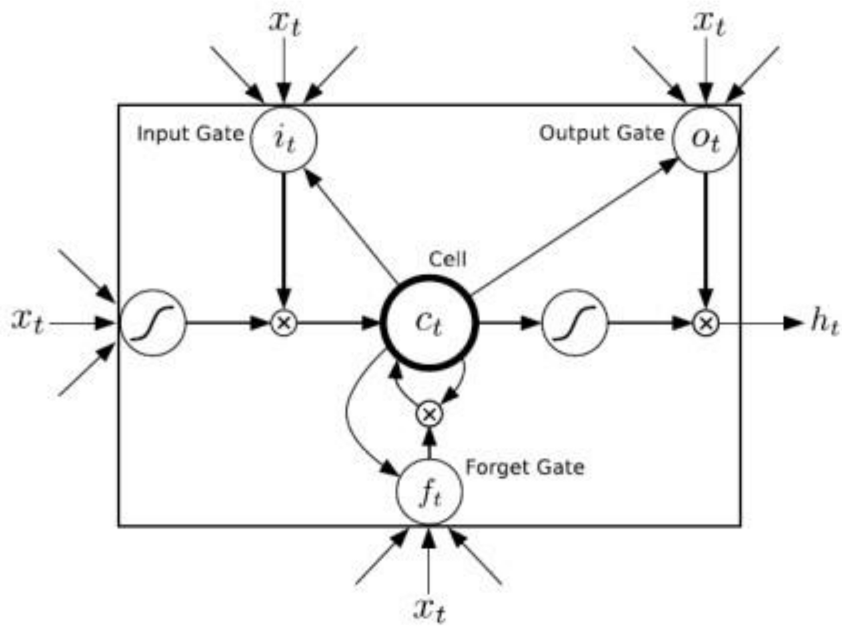

Figure 2. 1 Long Short-Term Memory cell (Nandanwar, 2021; Rengasamy et al., 2020)

Nandanwar, (2021)shows one cell of the LSTM memory block. More precisely, the input t x to the cells is multiplied by the activation of the input gate, the output to the net is multiplied by that of the output gate, and the previous cell values are multiplied by the forget gate. The net can only interact with the cells via the gates.

The LSTM units retain the prior state and retain the features that retrieved from the most recent data input. The LSTM variant Graves et al. reported adds weighted peephole connections from the Constant Error Carousel (CEC) to the gates of the same memory block, and this variant is the one used in this investigation. The peephole connections enable all gates to inspect into the current cell even when the output gate closed since they directly use the current cell state to create the gate degrees (Nandanwar, 2021).

The following components are composite of the LSTM-based recurrent neural networks (Zhou et al., 2016). The input gate it with a corresponding weight matrix Wxi, Whi, Wci, bi. The forget gate ft with a corresponding weight matrixWxf, Whf, Wcf, bf. The output gate ot with a corresponding weight matrix Wxo, Who, Wco, bo. All of those gates are set to generate some degrees, using the current input xi, the state $hi-1$ that previous step generated, and the current state of this cell $ci-1$ (peephole), for the decisions whether to take the inputs, forget the memory stored before, and output the state generated later. Just as these following equations, demonstrate (Zhou et al., 2016).

$it = \square\ (wxixt + whiht{-}1 + wcict{-}1 +)$ (2. 1)

$ft = \square\ (wxfxt + whfht{-}1 + wcfct{-}1 + bf)$ (2. 2)

$gt = \tanh\ (wxcxt + whcht{-}1 + wccct{-}1 + bc)$ (2. 3)

$ct = itgt + ftct{-}1$ (2. 4) $= \square$

$(wxoxt + whoxt{-}1 + wcoct + bo)$ (2. 5) $ht = ot$

$\tanh(ct)$ (2. 6)

Where $\square$the logistic sigmoid function and h is the hidden vector. Hence, the current cell state ct generated by calculating the weighted sum using both previous cell state and current information generated by the cell.

Bidirectional Long Short-Term Memory (BiLSTM)

Extended variants of unidirectional LSTM networks known as bidirectional long short-term memory recurrent neural networks. The fact that the crucial information can be anywhere in the sentence presents an unavoidable obstacle for relation extraction. Standard LSTM networks, on the other hand, process sequences in chronological order and neglect future context. The BLSTM network made to keep contextual elements from the past and future while capturing information from sequential data sets. In order to model the phrases, complete sequential information about all words before and after it employed in bi-directional LSTM networks (J. Lee, 2010).

Both the future and the past historical context are useful for the semantic relation extraction task. Because of this, we employ a BLSTM to access both future and historical context in order to obtain high-level characteristics.

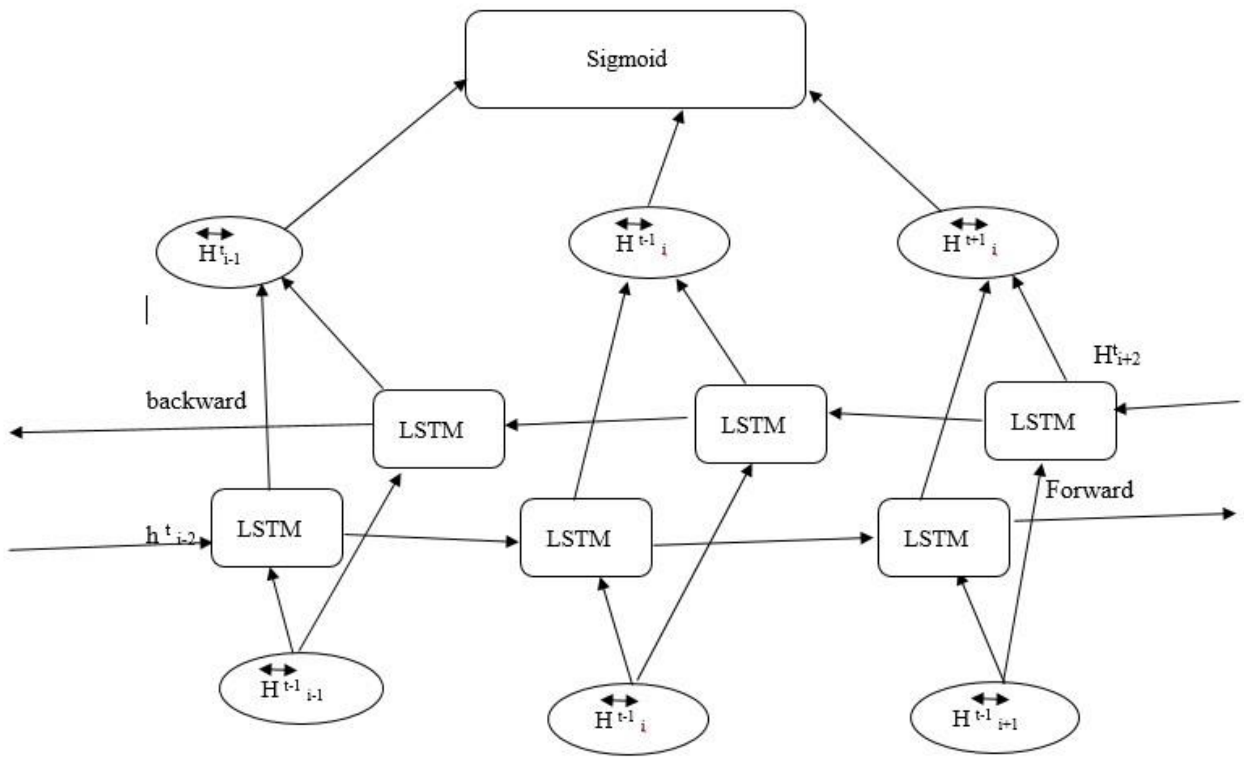

Figure 2. 2 Architecture of a Bidirectional Long Short-Term Memory Multi Label Relation Classification (Zhou et al., 2016)

$$ht = [ht^{\rightarrow} \oplus ht^{\leftarrow}] \quad (2.\ 7)$$

Here, we use the element wise sum ($\oplus$) to combine the forward and backward pass outputs. There are three motives for selecting BiLSTM for relation extraction. First, BiLSTM shows better performance. Second, LSTM-based models have meanings that are more explicit in the attention mechanism that is used in the next step than CNN-based models. Last, BiLSTM is quite simple compared with other complex models, which means it has fewer parameters and faster calculating speed.

Sigmoid Classifier

Sigmoid activation function is a type of logistic activation function. It used in the hidden layers of neural networks to transform the linear output into a nonlinear one. Softmax activation function used in the output layer of neural networks to convert the linear output into a probabilistic one.

Sigmoid activation functions used when the output of the neural network is continuous. Softmax activation functions used when the output of the neural network is categorical (deeplearning/what-is-the-difference-between-sigmoid-and-softmax-activation-function/, 2020).

2.6 Feature extraction

The term "feature extraction" refers to techniques for selecting or combining variables into features, which significantly reduces the amount of data that needs to process while properly and fully characterizing the initial data set. How features extracted is a significant distinction between deep learning and conventional machine-learning (Thankumar et al., 2019). Feature Extraction attempts to decrease the number of features in a dataset by generating new features from the ones that already exist (and then discarding the original features). Therefore, the majority of the data in the original set of features should be able to summarize by this new reduced set of characteristics. In this manner, a combination of the original set of features can result in a condensed version of the original features (Thankumar et al., 2019).

Traditional machine learning techniques utilize a number of feature extraction algorithms before using the learning algorithms to create handcrafted engineering features. A decision made based on the multiple outcomes from the various algorithms after applying several learning algorithms to the features of a single job or dataset in another common boosting strategy.

In the case of Deep learning, on the other hand, the features learned automatically and represented hierarchically at various levels (Benuwa et al., 2016; Igrejas et al., 2022; Voulodimos et al., 2018). This is where deep learning excels in comparison to more conventional machine learning techniques. The following is a description of the most typical features for the task of relation extraction using deep learning models.

2.7 Word embedding

The process of mapping words to vectors of real numbers known as word embedding. It is a more contemporary method for relation extraction using feature-learning algorithms. By learning from massive amounts of data, it helps to generate the semantic and syntactic similarities between words. Word embedding is work on the premise that any two words with a similar meaning will also have a comparable set of context terms. You may think of word embedding is as an unsupervised feature extraction method. As a result, it lessens the necessity for handcoding feature extractors and linguistic resources (Mikolov et al., n.d.).

Two approaches for learning word embeddings from raw text are word2vec (Mikolov et al., n.d.) and glove (Pennington et al., 2014). Both have demonstrated success in a variety of NLP tasks, including relation extraction. The word frequency and co-occurrence counts utilized as the primary metrics in the glove to capture the meaning of one word embedding with the structure of the entire observed corpus. Word2vec is a two-layer neural network that analyzes text by "vectorising" words to determine whether two keywords are similar. The continuous bag-ofwords (CBOW) model and the skip-gram model are the two main word2vec models. Let us talk about each of these models independently.

1) The Skip-Gram Model

When a word is given, the skip-gram model typically used to predict all adjacent words (context) (Mikolov, 2014). The representation dimension for skip-grams decreases from the size of the vocabulary to the depth of the hidden layer. The vectors also depict the link between words in a more meaningful way. According to the skip-gram paradigm, context words created based on the primary target word. The text sequence "ዘላለም", "በባህር", "ዳር", "ከተማ", "በ1985", "ዓ/ም" and "ተወለደ" is an example. The key word here is """ከተማ""".

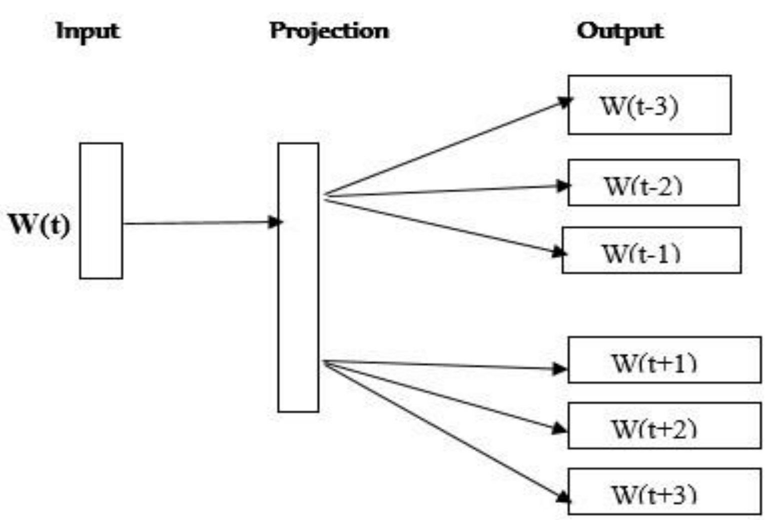

Figure 2. 3 the skip-gram model (Mikolov, 2014)

The model accepts a word W (t) and predicts the words around a given word W (t), which context words W (t-3), W (t-2), W (t-1), W (t+1), W (t+2), W (t+3). Within a predetermined window size, some words skipped to examine both ahead and behind the target word. There is only one projection layered neural network in the Skip-gram model. A vocabulary vector that has been one hot encoded makes up the input layer.

2) The continuous bag of words (CBOW) Model

To anticipate a word from context (the words around it), CBOW generates a sliding window around it (Mikolov, 2014). Any word or set of words can serve as the context. In the CBOW paradigm, the context words used to construct the core target word. The target word "ከተማ" can be produced using the CBOW model based on the context terms "ዘላለም", "በባህር", "ዳር", "በ1985", "ዓ/ም" and "ተወለደ".

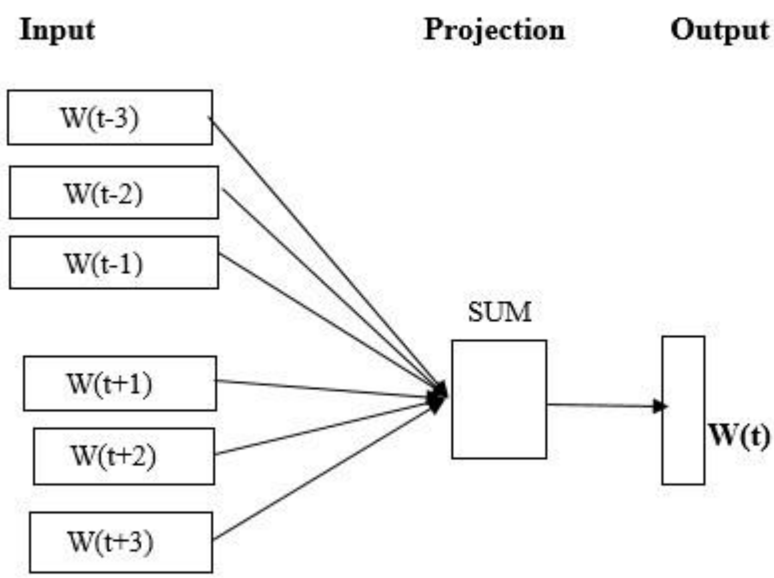

Figure 2. 4 The Continuous Bag of Words (CBOW) Model (Mikolov, 2014)

A continuous bag of words is the reverse of the skip-gram model. As shown in Figure 4 (Mikolov, 2014), the model accepts the context W (t-3), W(t-2), W(t-1), W(t+1), W(t+2), w(t+3) the task is to predict the target word W(t). The continuous bag of words model (CBOW) takes the average of the vectors of the input context words to compute the output of the production layer. As shown in Figure 3 and Figure 4, the difference between skip-gram and CBOW is the way the word vectors generated. In CBOW, the target word with all the examples are fed into the model and taking the average of the extracted hidden layer.

Given a sentence consisting of T Words $S = \{x_1, x_2, . . . \ x_T\}$, every word i x is converted into a real-valued vector e_i. For each word in S, we first lookup the embedding matrix $W^{wrd} \in R^{d\ w|V|}$, where V is a fixed-sized vocabulary and w d is the size of word embedding. The matrix W^{wrd} is a parameter to learned and w d is a hyper-parameter to chosen by the user. We transform a word X_i into its word embedding e_i by using the matrix-vector product (Mikolov et al., n.d.):

$$e_i = W^{wrd} v^t \tag{2. 11}$$

Where v^t is a vector of size |V | which has value 1 at index e_i and 0 in all other positions. The output of this phase is a fixed-sized vector representation for each word. Then the output sentence feed into the next layer as a real-valued vector. This output file will be the source of features in the next stages of this proposed architecture.

$$emb_s = \{1e, 2\ e, . . ., T\ e\} \tag{2. 12}$$

2.8 Position indicators

Position indicators are required in relation extraction to inform the algorithm about the target nominal (Qin et al., 2021). Position indicators (PI) are crucial for improving the precision of relation classification. The input word sequence S, for instance, shown below along with minimum four and maximum eight position indications (PI) that indicate the beginning and end of the nominal. The following is an Example (Qin et al., 2021):

"አቦ <e1>በቀለ ገርባ</e1> <e2>የኤልቲቪ</e2> ባለቤትና መስራች ጋር የተደረገ ልዩ ቆይታ ።" In this sentence the position indicators are only four; these are,

'<e1>': a word before the first relation argument in the word sequence

'</e1>': a word after the first relation argument in the word sequence

'<e2>': a word before the second relation argument in the word sequence

'</e2>': a word after the second relation argument in the word sequence

"<e1>በመሰለ ገብረህየወት</e1> የተጻፈው <e2>የሚዲያ አመራር</e2> የተባለው መጽሐፍ እሁድ ሐምሌ 3 በተወለደባት ታሪካዊ በሆነችው <e3>ደሴ ከተማ</e3> ይመረቃል ።" In this sentence the position indicators are six such as,

'<e1>': a word before the first relation argument in the word sequence

'</e1>': a word after the first relation argument in the word sequence

'<e2>': a word before the second relation argument in the word sequence

'</e2>': a word after the second relation argument in the word sequence

'<e3>': a word before the third relation argument in the word sequence

'</e3>': a word after the third relation argument in the word sequence

"<e1>የአርበኞች ግንቦት 7</e2> ሊቀመንበር ፐ/ር <e2>ብርሃኑ ነጋ</e2> <e3>በኦስሎ</e3> ፣ <e4>ኖርዌይ</e4> የኢትዮጵያ ብሔራዊ ንቅናቄ ህዝባዊ ስብሰባ ላይ ያደረጉት ንግግር ።"

'<e1>': a word before the first relation argument in the word sequence

'</e1>': a word after the first relation argument in the word sequence

'<e2>': a word before the second relation argument in the word sequence

'</e2>': a word after the second relation argument in the word sequence

'<e3>': a word before the third relation argument in the word sequence

'</e3>': a word after the third relation argument in the word sequence

'<e4>': a word before the fourth relation argument in the word sequence

'</e4>': a word after the fourth relation argument in the word sequence

2.9 Evaluation metrics

By testing the model on the recently created Amharic-RE-Dataset for the relation extraction task, the proposed system's efficacy is shown. Precision, recall, and the F1-Score are the evaluation measures to assess relation extraction methods. Precision is the ratio of successfully retrieved relationships of type r over all successfully retrieved relationships of type r in the text. Recall is the percentage of successfully extracted relationships of type r over all relationships of type r present in the text (Martin & Powers, 2015).

An attempt to combine these measures is the F1-Score, which corresponds to the harmonic mean of both precision and recall. Where r refers to a type of relationship, and N refers to the number of relationship type's r.

True Positives (TPr) is the number of successfully extracted relationships of type r;

False Positives (FPr) is the number of extracted relationships that said to be of type r but are not from type r.

True Negatives (TNr) is the number of successfully extracted relationships that were not of type r.

False Negatives (FNr) is the number of extracted relationships that are of type r but said to be other type. Accordingly, Precision, Recall, and F1-Score defined as (Martin & Powers, 2015):

$$\text{Precision (Pr)} = \frac{\text{Number\_of\_correctly\_extracted\_relations r}}{\text{total\_Number\_of\_extracted\_relations r}} = \frac{TPr}{TPr+FPr} \tag{2.13}$$

$$\text{Recall (Rr)} = \frac{\text{Number\_of\_correctly\_extracted\_relations r}}{\text{Total\_Number\_of\_Relations\_types r in\_Tex r}} = \frac{TPr}{TPr+FNr} \tag{2.14}$$

$$\text{F1-Score (F1r)} = 2 * \frac{Pr * Rr}{Pr+Rr} \tag{2.15}$$

These metrics are automatically defined per label, however when dealing with multi-label classification issues, it can be helpful to expand them in order to create a final score.

2.10 Data source

To execute analysis, the relationship extraction method requires enough data. RE systems cannot achieve good performance if the data source is not large enough or representative of the domain. Typically, different approaches are used on different types of data by RE systems. The accuracy of the training data is also important for the improved performance of the relation extraction system. Three types of data sources can be found in relation extraction: free text (unstructured), semi-structured text, and structured text (Adnan & Akbar, 2019; Sivarajah et al., 2016).

A) Free or Unstructured text

Unstructured text (free text) is just narrative text with no intentional formatting. Newswire reports, newspapers, journal articles, electronic communications, and other sources may be used. The goal of entity relation extraction was originally to create practical systems that could take short natural language texts and extract a limited set of critical pieces of information. For example, the texts could include news stories on terrorist acts, with essential information like as the perpetrators' affiliations, the victims' whereabouts, and so forth. Managing free text is difficult due to the lack of a well-defined framework.

Natural language approaches frequently utilized in entity RE systems free text, and extraction criteria typically based on patterns involving syntactic interactions between words or semantic classes of words. Syntactic analysis, semantic tagging, and recognizers for domain objects such as person and company names, and extraction rules are all required.

The rules or patterns can be hand-coded or produced from training examples labeled by a human expert with the appropriate label. The current state of the art in free text information extraction is not comparable to human capabilities, yet it still produces meaningful results. This is true regardless of whether the rules manually coded or learned automatically. Because narrative material is generally quite complicated, automatic entity RE systems perform poorly when compared to hand-coded solutions for unstructured text. However, RE systems can still produce meaningful results on narrative text, owing to the fact that they rely on specific elements with a predictable structure.

B) Semi structured text

Semi structured data are a middle ground between unstructured collections of textual texts and fully structured typed data tuples. Entity RE systems have historically been unable to access such texts because they fall between structured and free text. Semi structured text is ungrammatical and frequently telegraphic in style, and it does not adhere to any strict format. Semi structured material does not always contain complete sentences.

Semi-structured text has a format in some ways, although the format's structure is less precise than that of structured text. To build rules for extracting information from free text, a variety of NLP techniques used. These strategies, which are appropriate for grammatical language, will almost never work for semi-structured material, which rarely comprises whole sentences. As a result, typical entity RE techniques not utilized for semi-structured texts, and basic rules employed for rigorously structured texts will not suffice.

C) Structured Text

Textual information in a database or file that follows a predefined and rigid format referred to as structured text. Using the format description, such information simply extracted. For extracting information from text, basic procedures are usually sufficient if the format known; otherwise, the format learnt. When compared to free text or semi-structured text sources, structured texts given in a table or database schema, making it easier to extract the relevant one. Because structured text given in a database schema is easier for a machine to grasp, entity RE research involving structured text is less common than research involving unstructured and semi-structured language.

2.11 Overview of Amharic Language

Amharic is the official working language of Ethiopia's Federal Democratic Republic and, after Arabic, the world's second most widely spoken Semitic language (Date, 1995). Amhara, Addis Ababa, South Nation Nationalities and People, Dir Dawa, and other regional states use it as a working language. It also taught in primary and secondary schools around the country. Around a 32 million speak it as their primary language as of 2018, plus 26 million people speaks it as a second language. Different forms of mass media, such as radio, television broadcasts, and the press, are currently employing it to disseminate information to the general audience. The language has few computational linguistic resources, despite its enormous speaker community.

Amharic, like other Semitic languages, is a morphologically complicated language.

2.11.1 Amharic writing system

Amharic is written using the Fidel writing system, which developed from the Ge'ez language. The Amharic writing system uses the entire Geez alphabet, including superfluous characters. For example, ሐ, ሀ, ኀ and ኸ are pronounced as (hä), ሰ and ሠ as (sä), አ and ዐ as (a) and, ጸ and ፀ as (tsä) (Salawu & Aseres, 2015). There are no capital and lower case letters in the Amharic alphabet. Amharic differs from Semitic languages in structure, particularly in syntax. It is written in a seven-column tabular format for ease of use. The first column depicts the basic form, from which the remainder generated through simple adjustments. Amharic contains 435 characters with 34 base characters. A single symbol or character in Amharic indicates both a consonant and a vowel. The six orders created by combining consonants and vowels using diacritic markings on the base symbol. The Ethiopic script is a syllabary rather than an alphabet since it lacks separate vowel letters, unlike alphabets in other languages, yet it is called an alphabet for convenience (Asker et al., n.d.).

2.11.2 Amharic Punctuation mark and Numerals

Amharic has its own punctuation system. In handwriting, there are many different punctuation marks. The most commonly used punctuation is (*Addis Ababa, Ethiopia March 2020*, 2020; Ashagrie & Boran, 2019):

- Colon (:), which is referred to in Amharic as "ሁለት ነጥብ." The two dots now replaced with whitespace.
- Four dots (።): also referred as አራት ነጥብ, used as a sentence delimiter, like the period symbol "." in the English equivalent
- Netela serez, ነጠላ ሰረዝ (፣) used to separate lists or concepts.
- Dirib serez, ድርብ ሰረዝ (፤): serves the same function as the semi-colon in English. In addition to the punctuation symbols described above, the language uses? /, ", ", ', and other punctuation marks borrowed from other languages.

Two numeral systems employed in the Amharic writing system. The first derived from the Geez language. However, because there is no sign for zero in this numbering system, these numbers are unsuitable for mathematical computation. It primarily used for page numbers and dates. The English numbering system is the second numbering system. This one is better suited to applications that require automatic Amharic document processing.

2.11.3 Amharic Sentence

One of the most morphologically complicated languages is Amharic (Aschenaki Abi Abera, Yaregal Assabe , Mesfin Kifle , 2020). Different affixes utilized to create inflectional and derivational morphemes in this language. Affixation (prefix, infix, and suffix) or compounding used to achieve the derivation. By modifying vowels or repeating consonants, and then adding the relevant affixes or suffixes, inflection created. In Amharic sentences, the word order is usually subject object verb (SOV) (Aschenaki Abi Abera, Yaregal Assabe, Mesfin Kifle, 2020).

A noun phrase and a verb phrase make up the grammatical framework of an Amharic sentence. The noun phrase placed first, followed by the verb phrase. By implicitly merging the object, subject, and verb in Amharic, one word can form a sentence. For example, the word "አላመጣላቸውም" interpreted as a sentence with the subject "እሱ" the object "እነሱ" and the verb "አላመጣም" As a result, recognizing morphemes from a word is challenging. Apart from multiple nouns, plural verb construction in Amharic differs from that in English. Amharic words are divided into five categories, according to (Salawu & Aseres, 2015). The use of morphology and the position of the word in a sentence used to classify these words. The noun, verb, adjective, adverb, and preposition are the word categories in Amharic.

In this thesis, the relation extraction model based on Amharic sentence structures. In Amharic sentences, the types of relations discussed in the following (*Addis Ababa, Ethiopia March 2020*, 2020):

Example 1: አስር አለቃ <e1>በየነ</e1> <e2>የኢትዮጵያ አየር ሀይል</e2> አባል ነው። this sentence is a type of relation falls in አባል (e1, e2) relation. Where e1 refers to በየነ (person) and e2 refers to አየር ሀይል (group or organization). In this case, በየነ (the person) is member, and አየር ሀይል (an organization) is a collection. Thus, a collection must enclose the person.

Example 2: <e1>ባህር ዳር ዩኒቨርሲቲ</e1> </e2>በባህር ዳር ከተማ</e2> ይገኛል። this sentence is a type of relation falls in መዳረሻ (e1, e2) and የለም between (e1, e2) predefined relation type. In this case, ባህር ዳር ዩኒቨርሲቲ (Institution) is an entity, and በባህር ዳር (city) is a destination. An entity ባህር ዳር ዩኒቨርሲቲ is located or destined at ባህር ዳር city. የለም (e1, e2) which means the sentence has neither triple nor multiple relation but it has only one relation between the mentioned named entities.

Example 3: <e1>በ1987 ዓ/ም </e1> የተወለደው </e2>ቢኒያም</e2> <e3>በባህር ዳር ዩኒቨርሲቲ</e3> መምህር ሆነ. This sentence is a type of relation falls in የተወለዱበት_ዘመን between (e1, e2) and የስራ_ቦት between (e2, e3) predefined relation types. In this case, በ1987 ዓ/ም is a time, ቢኒያም is a person and ባህር ዳር ዩኒቨርሲቲ is (Institution) is an entity. An entity በ1987 ዓ/ም is date of birth to ቢኒያም isworks at ባህር ዳር ዩኒቨርሲቲ.

Example 4: የሎሬት <e1>ፀጋዬ ገብረመድህን</e1> ከሕይወት <e2>ቢራቢሮ</e2> መታተም ጋራ ተያይዞ ለአንድምታ ባልደረባ ያወጋው ነው ። these sentences are a type of relation falls ጸሀፊው_ነው between (e1, e2) and የለም (e1, e2) predefined relation types. Then ፀጋዬ ገብረመድህን is a person writes a book named ቢራቢሮ.

Example 5: <e1>የሶማሌ ክልላዊ መንግሥት</e1> መዲና <e2>ጅጅጋ ከተማ</e2> የተወለደው አርቲስት <e3>ዝናብዙ</e3> ከዚህ አለም በሞት ተለዩ ። these sentences are a type of relation falls two relation type known as ዋና_ከተማ between (e1, e2) and የተወለዱበት_ቦታ between (e2, e3) their predefined relation types. Then >የሶማሌ ክልላዊ መንግሥት is organization is capitals is ጅጅጋ ከተማ and a city birthplace of artist ዝናብዙ.

There is a predefined relation type called ሌሎች, which means they are a relation between the named entities. However, it is out of our mentioned predefined relation type.

There is a preset relation type, which denotes that there is just one relation or no triple relation between the named entities mentioned in the sentence. E.g. <e1>ዘላለም</e1> <e2> በቡሬ ከተማ</e2> ተወለደ ። When we investigated this statement, we discovered that it comprises two named things, a PER with LOC, and just one relation between them, known as የተወለዱበት_ቦታ. For better understanding triple and multiple relation, the former is a relation which exists two or more relations between two named entities and the later a sentence have more than two entities then the relation between the mentioned entities is more than one relation exist.

Consider the following example:

Triple relation ዘላለም በባህር ዳር ዩኒቨርሲቲ ተመርቆ መምህር ሆነ ። the triple relation falls between ዘላለም and በባህር ዳር ዩኒቨርሲቲ የተመረቁበተ_ትምህርት_ቤት and የስራ_ቦታ

Multiple relation የሶማሌ ክልላዊ መንግሥት መዲና ጅጅጋ ከተማ የተወለደው አርቲስትን ዝናብዙ ከዚህ አለም በሞት ተለዩ ። the relation falls between ዋና_ከተማ(የሶማሌ ክልላዊ መንግሥት, ጅጅጋ ከተማ) and የተወለዱበት_ቦታ (ጅጅጋ ከተማ, ዝናብዙ)

2.12 Related works

In this section, we present some of the entity relation extracting work done thus far. Among them, we have selected the most pertinent ones that related to our work in different languages.

2.12.3 Entity Relation Extraction from Arabic Text

Because Arabic is a highly inflectional and derivational language, it has complex morphological, grammatical, and semantic aspects, making the work much more difficult (AlArfaj, 2019). Methods for extracting semantic entity relations classified as supervised or unsupervised depending on the learning paradigm they use. The goal of supervised techniques is to figure out which types of relationships exist between entities by employing predefined relationships. Support Vector Machine, Conditional Random Fields, and Maximum Entropy are some of the machine learning techniques that used to extract relationships.

Supervised approaches necessitate annotated training data and specified relationships. They developed a semi-supervised method for learning semantic relationships between named things that uses labeled and unlabeled relation instances. There are three basic methods for extracting taxonomic relationships from text. The lexico syntactic patterns, such as Hearst patterns, are the earliest. Although this method has a great level of precision, it has a relatively low recall. As a result, these patterns are uncommon in the corpus. Therefore, to fix the problem there must be more corpora prepared to find more patterns.

The second method based on the Harris distributional hypothesis, and it involves extracting idea hierarchies from text using hierarchal clustering techniques. it performs two tasks using clustering approaches: concept formation and concept hierarchy induction (Wu et al., 2016).

The third technique is based on the assumption that the presence of some words in a sentence, paragraph, or document suggests the presence of other words in the same sentence, paragraph, or document, indicating relationships between the two words (Baader et al., 2004). To label relations and concept clusters, the statistical-based technique requires user interaction during the validation step. However, compared to lexico-syntactic approaches, which require an expert for pattern preparation and building, this approach requires less preparatory data. Ontology construction requires the extraction of relationships.

The majority of known methods concentrate on extracting taxonomic relationships. There have only been a few ways to learning non-taxonomic relationships from text. Except for the IS-A relationship, non-taxonomic interactions are relationships between idea pairs. For example, in the meronomic relation (part-whole or part-of relation) that exists between two conceptions where one concept is a part of the other.

Al Zamil & Al-Radaideh,(2014) presented a pattern-based and seed ontology method for extracting antonyms from an Arabic corpus. The extracted patterns used to find new antonym pairs to add to the ontology. Their evaluation results showed that the system enriched ontology with 400% increase in size. For extracting new antonyms, their result showed that only 2.7% of the patterns were useful. (Boudabous & Chaâben, 2013) Based on Wikipedia, a hybrid strategy for building Arabic ontologies presents. They improved AWN (Arabic WordNet) by introducing semantic links between synonymy sets using a linguistic technique based on morpho lexical patterns. They define morpho lexical patterns first, and then use them to enrich semantic relationships.

Boujelben et al.,(2014) proposed the relationship between Arabic NEs, which studied using a hybrid system that combines the benefits of machine learning and rule-based approaches. ML used first, followed by a rule-based post-processing technique. Is employed in order to improve the overall performance of the machine-learning system the goal is to anticipate the trigger words that will be used. Elucidate the semantic relationships between NEs in Arabic Text Using a set of rules as a guide. To begin, the approach use machine learning to extract rules via a decision tree technique and an Apriori algorithm.

The most significant and interesting rules are then extracted and generated using a genetic algorithm (GA). Following the application of machine learning approaches, we incorporated handcrafted rules to deal with incorrect examples and unknown relationships. The author's goal is to find the word's location that reflects a meaningful relationship between NEs. To improve the overall performance of the ML method, we merged the ML method for automatically extracting rules based on the GA with some language modules. In addition, the authors restrict extracting their relation between NE such as PERS, ORG and LOC.

2.12.4 Entity Relationship Extraction from Chinese text

Traditional methods of relationship extraction, whether was proposed earlier or based on traditional machine learning and deep learning, have focused on keeping relationships and entities in their own silos: extracting relationships and entities in steps before obtaining mappings this solution cannot efficiently deal with the entity relationship extraction entity overlap, relationship crossover, and so on. Therefore**, (**C. Lv et al., 2021) proposed a novel Chinese relationship extraction method to overcome this issue.

Firstly, the Bidirectional Maximum Entropy Markov integrated into the Joint Extraction of Entity Mentions and Relations model, which is similar to Seq2Seq. Second, unlike other connection extraction techniques, relationship triples handled as an entity relationship chain, with entity E1 being detected first, followed by the related relationship R and entity E2 based on E1. Finally, the proposed model's validity is tested using Chinese data sets, and its scalability assessed using English data sets. The Bi-LSTM Layer and Tanh Layer/Attention translate character-word-position embedding into coding matrix M.

Secondly, In the Bi-MEMM Layer and Dense Layer, matrix M copied. The Dense Layer can use Sigmoid as an activation function. The head and tail positions of E1 therefore be predicted using a two-dimensional vector generated by each letter.

Thirdly, the subsequence corresponding to E1 is fed in M into the first Self Attention Layer, along with the Position Embedding at the corresponding position, and it is transformed into a vector with the same length as the input sentence (randomly pick E1 when training, and traverse all E1's when predicting).

Finally, matrix M sent through the Bi-MEMM Layer and the Dense Layer. The Dense Layer with the activation function of may also anticipate the head and tail positions of E2 for each R corresponding to E1. The proposed model has achieved a precision of 79.2%, which is much higher than as compared to the traditional approach.

2.12.5 Distant supervision for relation extraction without labeled data from French text

Doshi (2018) present a modified version of Deepdive for French, which can be interesting for the application of non-English languages. Deepdive's Architecture consists of three phases, feature extraction, probabilistic engineering, and statistical inference and learning. Deepdive gets linguistic features by using tools like named-entity recognizer and dependency pathfinder. Then these features used to discover correlations between linguistic patterns and relations defined by the user.

In this case, we developed an Amharic relationship extractor as a preliminary study. Xu et al. (2022), process an event information integration model that uses a multilayer bidirectional long short-term memory (Bi-LSTM) and attention mechanism to integrate event data. In the meanwhile, the aforementioned strategy can increase extraction performance, but it can still improve. We present a unique relational graph attention network that integrates edge properties to improve the performance of the previous system. We use dependency parsing to build a semantic dependency graph, and then use top-k attention techniques to learn hidden semantic contextual representations to model a semantic network that analyzes the edges' properties, and lastly forecast event temporal relations.

Finally, the proposed model out performs the entire previous model and can improve the f- score by 3.9% by experiment Time Bank-Dense dataset. Lv et al. (2021) proposed a model has entity relation chain to identify the head entity before relationship should be firstly, and then, the corresponding relationship and the tail entity predicted. For instance, in the sentence "አቶ ዘላለም በባህር ዳር ከተማ ተወለደ," E1 "አቶ ዘላለም" and E2 "በባህር ዳር ከተማ" usually identified firstly and the R "ተወለደ" recognized secondly. However, in the entity relation chain, E1 "ዘላለም" firstly identified, and every possible R generated from E1 is the criterion for E2 "ባህር ዳር ከተማ". In this entity relation chain, E1 is taken as head entity, R as relation chain, and E2 as tail entity.

The result has a highly performed. The proposed model can achieve a precision of 79.2%, which is much higher than that of traditional models. The proposed approach did not say anything about multiple relations in a sentence and co-occurrence between sentences. Qin et al. (2021), conducted a study on Task-related entity indicators are designed to allow a deep neural network to focus on the task-relevant information rather than learning an abstract representation from raw inputs. The neural network is effective in encoding syntactic and semantic information about a relation instance by implanting entity indicators into it. Entity indicators that are organized, structured, and unified can make the resemblance between sentences with the same or comparable entity pair, as well as the internal symmetry of one sentence, more visible.

The authors specify three types of entity indicators are proposed: position indicators, semantic indicators and compound indicators. Position Indicators point to the positions of two arguments in a relation mention. Semantic Indicators specify entity type and subtype contain important semantic information about named entities. Compound Indicators: The above semantic indicators have shown the ability to combine semantic information and positional information.

The proposed method achieved the highest precision is 76.24% on NYT-FB (New York Times - Freebase) dataset (top 100 relation categories). Huminski & Bin, (2020), proposed for causal chain extraction, a new approach based on language templates. It is domain-agnostic, not limited to single-sentence extraction, and can be unfurled on large datasets. A four-module sequence used to implement the system.

Verb limitation, part-of-speech labeling, causal links extraction, and unification and matching events are among them. However, it cannot fix wrong label problem due to a sentence it does not has a relation. Peng & Chen, (2020), attempted to undertake two main steps. In the first step, causal relations found by matching pre-defined linguistic templates. In the second step, causal chains are constructed by joining the relations using the process of unification and matching and the same exact string matching is insufficient for extracting casual relation, due to bad POS tagging the casual relation extraction failed.

Zhong & Chen, (2021), proposed a system that work in a way similar to how human reader processes a story to understand the main characters and their relationships. Sometimes these are directly given in story, but most of the time the reader has to infer the relationships by analyzing conversations, some statements, contexts etc. the proposed system 1.7%- 2.8% absolute improvement in relation extraction F1 over previous joint models with the same pre-trained encoders. The paper does not say the anaphora resolution, co-reference and complex relationship within a sentence. Yadav et al. (2020), proposed two independent models which are entity model and relation model trained independently and the relation model only relies on the entity model to provide input features. Entity model builds on span-level representations and our relation model builds on contextual representations specific to a given pair of spans. Entity model predicts all the entities at once. Relation model considers every pair of entities independently by inserting typed entity markers.

Li et al. (2020), designed a multi-turn question answering paradigm. It has two major stages these are the head-entity extraction and relation and the tail extraction. Each episode of multi-turn QA is triggered by an entity. ChainOfRelTemplates defines a chain of relations, the order of which we need to follow to run multi-turn QA. The reason is that the extraction of some entities depends on the extraction of others. The extraction of entities and relations transformed to the task of identifying answer spans from the context. This multi-turn QA formalization comes with several key advantages: firstly, the question query encodes important information for the entity/relation class we want to identify; secondly, QA provides a natural way of jointly modeling entity and relation; and thirdly, it allows us to exploit the well-developed machine reading comprehension (MRC) models.

Experiments on the ACE and the CoNLL04 corpora demonstrate that the designed paradigm significantly outperforms previous best models. Yadav et al. (2020), attempted to model the relation extraction problem in a multi-task learning (MTL) framework, and introduce for the first time the concept of structured self- attentive network complemented with the adversarial learning approach for the prediction of relationships from the biomedical and clinical text. The fundamental notion of MTL is simultaneously learn multiple problems together by utilizing the concepts of the shared representation. Additionally, we also generate the highly efficient single task model, which exploits the shortest dependency path embedding learned over the attentive gated recurrent unit to compare our proposed MTL models.

The proposed model leverages joint modeling of the entities and relations in a single model by exploiting attentive Bi-GRU based recurrent architecture. The authors propose also an adversarial multi-task learning with attention (Ad-MTL) model for relation extraction task. The proposed model outperforms and superior to the state of the art. However, it does not concern multiple relation in a given sentence and co-occurrence between sentences. Agosti et al. (2019), suggested semantic relations for case-based retrieval. Relation extraction and retrieval are two components of this technology. An entity-connecting component and a relation extraction component make up the relation extraction stage. The entity-linking component takes entity mentions from the text and links them to a reference KB, reducing the number of synonyms, abbreviations, and context-specific terms seen in medical literature. Within a sentence, the relation extraction component discovers semantic relationships between pairs of ideas. Lee et al. (2018), proposed dynamic convolutions based on lightweight convolutions to process long sequences, which thus reduces the number of parameters to a low level.

To extract entities and relations from unstructured texts, the authors use reinforcement learning and deep learning. Feng et al. (2017) proposed model the problem as a two-step decision process for reinforcement learning. Deep learning used automatically extracts the most significant information from unstructured texts that indicate a decision's status. Our suggested technique can transfer entity extraction information to relation extraction and collect feedback in order to extract entities and relations simultaneously by constructing the reward function per step. To begin, we model the context information using bidirectional LSTM, which allows us to do preliminary entity extraction.

The attention-based method can represent the sentences that include the target entity pair to construct the starting state in the decision process based on the extraction findings. Then, to construct the transition state in the decision process, we employ Tree-LSTM to represent relation mentions. Finally, in the two-step decision process, we use the Q-Learning method to obtain control policy. Finally, Experiments on ACE2005 show that our method outperforms the state of-the-art method, resulting in a 2.4% gain in recall-score.

Another challenge in relation extraction is relation spans potentially overlapping in a sentence, representing a bottleneck for the detection of multiple relational triplets. To alleviate this problem, we design an entirely new prediction scheme to extract relational pairs and additionally boost performance. The proposed model has two parts these are E1 prediction and multi-turn E2 prediction. Firstly, the encoder converts the input sentence into a fixed-length vector where a 12layer GLDR and dynamic convolutions are used. In this step, we need to extract all of the E1s of the sentence and place them into a “bag”.

Then we sample an E1 from the bag and encode it with a BiLSTM layer. This side information is used to us predicting E2 and the relation between them. Devisree & Raj, (2016), proposed a system for any sentence that contains a form of [s nsubj ←h nmod: prep → o] where s is an entity subject, prep is some preposition, o is an entity object, and h is the common headwords, then we can extract from the sentence. A relation < s concat (h, prep) o > where concat is the string concatenation function.

Relations extracted by this rule, for example, can include the predicates livedIn, marriedTo, etc. The main applications are story summarization and analysis of the major characters in stories. The proposed system gives an average Precision of 87% and an average Recall of 79.7%. The results show that proposed approach performed well with the given input text.The proposed system talks only two entities within a sentence and cannot say anything about co-references.

Prasojo (2016), proposed model for relation extraction comprises three parts: preprocessing, convolutional neural network (CNN), and post processing. Firstly, takes as input each raw text (i.e., a paragraph of a scientific article in ScienceIE) as well as the location of all entities present in the text, and output several examples. Secondly, takes each preprocessed sentence as input, and predicts the relation between the two entities. In addition, the third step to correct the relations detected by CNN is to detect additional relations.

These rules developed from the examples in the training set, to be consistent with common sense. X. Lv et al. (2016), proposed CRF-powered classification model depends on features of context of concepts. To remedy the problem of word sparsity, a deep learning model applied for features optimization by the employment of auto encoder and sparsity limitation. The proposed model designed based on CRF. Specifically, CRF++ adopted, which is the mostly applied implementation of CRF model. The proposed model validated on the data set of I2B2 2010. The experiments give the evidence that the proposed model is effective and the method of features optimization with the deep learning model shows the great potential.

Muzaffar et al. (2015), proposed five major modules of relation extraction framework: corpus-preprocessing, natural language processing, UMLS based ranking of noun and verb phrase, creation of n-dimensional vector space, and classification of entities. The authors used the supervised learning methods that used SVM and NB classifier to evaluate our feature set. Approach validated on the standard biomedical text corpus obtained from MEDLINE 2001. Conclusively, it articulated that the proposed framework outperforms all state-of-the-art approaches used for relation extraction on the same corpus. The proposed approach did not say about multiple relations within a given sentence and co- references across a sentence.

2.12.6 Semantic relation extraction from Amharic language

Asch et al, (2020) suggest deep learning approach for extraction the relation between named entities from Amharic language texts. The suggested model consists of various parts. A word embedding converts each word into a low dimension vector as the first option. In order to extract relationships from Amharic text, feature-learning techniques used to acquire new features from other domains. The second is BiLSTM, which assists in obtaining high-level features from the embedding layer by utilizing data those points in both the past and the future. Not all-contextual information reflected by a single direction of relationship. The third attention mechanism creates a weight vector and, by multiplying it, integrates word-level characteristics from each time step into a sentence-level feature vector.

The authors also used word embedding approach (word2vec) used to map each word into a low dimension vector for automatic feature generation after the unlabeled free text has been preprocessed. Before being used for training, the enormous amount of unlabeled input data will be tokenized and converted into vectors by seeking upward embeddings. By connecting the many conceptual terms used in a text, we can deduce the contexts of the words in a text from the word2vec result.

Authors used to test the suggested approach the Amharic-RE-Dataset, which created from Amharic text. The efficacy of the suggested approach is evaluated using the evaluation methods recall, precision, and F-score. Finally, the suggested attention based bidirectional long short-term memory model responds an F-score of 87.6%. When we investigated the sentence, it may have more than two named entities because of this the relation in between also possibly triple or multiple. In general, the suggested approach works based on two named entities that has only one relation within a given sentence between the mentioned named entities.

To summarize, many works that published in the field and have attempted to fix a link between named entities taken and thought to have only one relation within a sentence. However, when we studied the sentence, there might have been one or more relations between them, and some sentences had triple relations. For example, ፀጋዬ ገብረመድህን 1928 ዓ/ም ቦዳ ተብላ በምትታውቅ አምቦ ከተማ አካባቢ በምትገኝ ተራራማ ቦታ ተወለዱ ። when we see this sentence, it contains three named entities called PER, LOC, TIME and multiple relations between them with (PER, LOC) ፀጋዬ ገብረመድህን, ቦዳ and (PER, TIME) ፀጋዬ ገብረመድህን, 1928 ዓ/ም then after it has a predefined relation type የተወለዱበት_ቦታ and የተወለዱበት_ዘመን respectively. Another e.g. ጋዜጠኛ መዓዛ ሞሀመድ የሮሃ ሚዲያ መስራች እና ባለቤት ናት ። this sentence contains only two named entities called መዓዛ ሞሀመድ , መዓዛ ሞሀመድ and has triple relation between መዓዛ ሞሀመድ and መዓዛ ሞሀመድ finally it has a predefined relation type መስራች and ባለቤት exists. As a result, our research focuses on fixes these triple and multiple relations within a sentence starts with the preparation of a dataset that is suitable for machine learning algorithms and ends with experiments.

CHAPTER THREE፦

METHODOLOGY

3.1 Overview

The proposed model for the Amharic semantic entities relation extraction system employing Bidirectional Long Short-Term Memory (BiLSTM) discussed in this part. The suggested method seeks to automate the extraction of entity relations from texts published in the Amharic language by automatically learning features from them. Details of the model discussed in this chapter. First, the proposed architecture presented. This followed by a discussion of text pre-processing tasks such as cleaning, sentence tokenization, tokenization, and stop-word removal. Finally, each component's interconnection and utilization detailed in its own subsection.

3.2 The Proposed architecture

In this section, the Amharic semantic entities relationship extractor model designed, as shown in Figure 3.1. As a deep learning strategy, Bi-directional Long Short Memory (BiLSTM) model is used. To capture both long-term dependencies and local features, the proposed model employs feature extraction, BiLSTM. Different components make up the suggested model. Preprocessing, word embedding, feature extractor, model builder, and relation extractor are the tools available. The following sections provide a quick overview of these components.

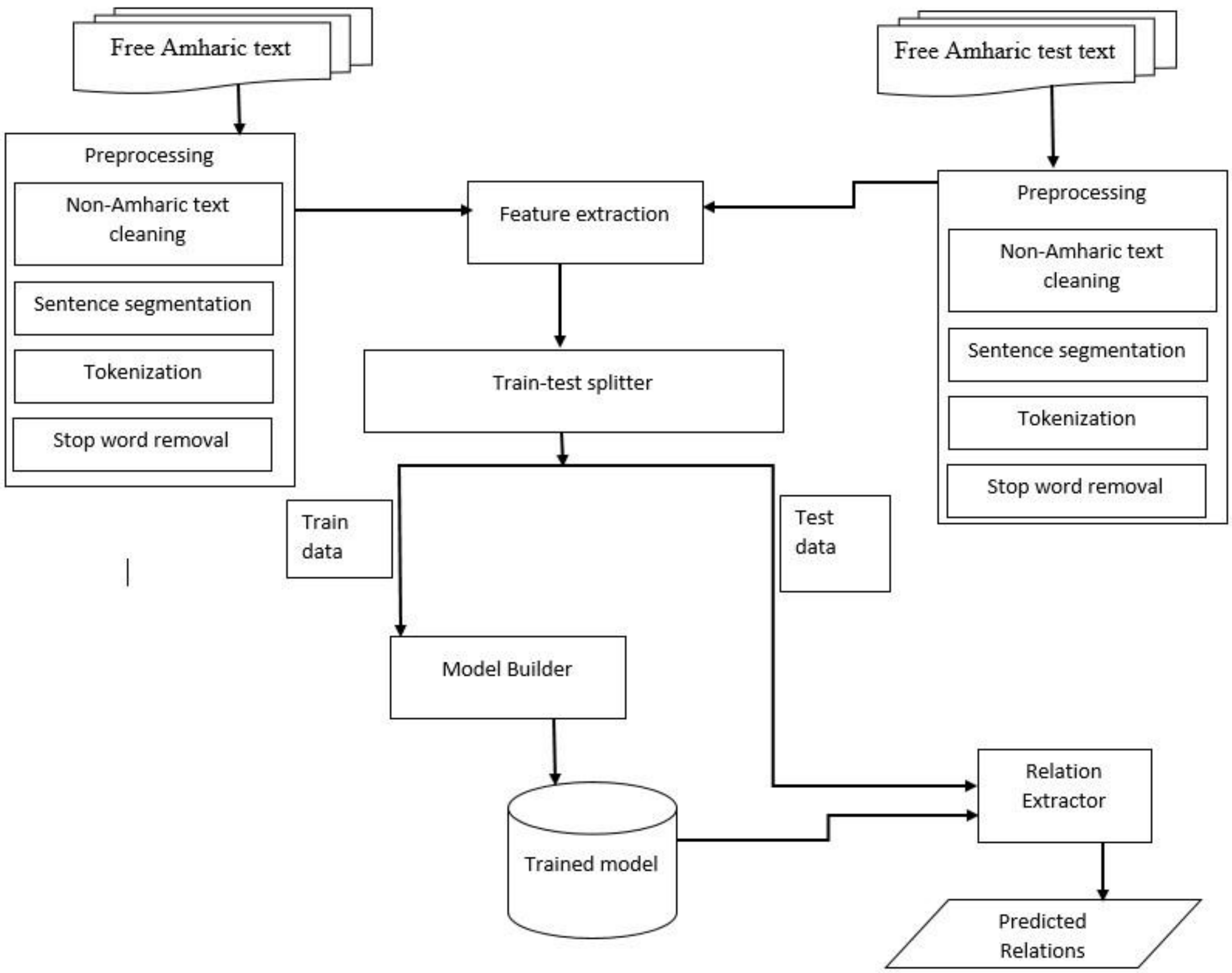

Figure 3. 1 Proposed System Architecture

3.2.1 Preprocessing

The input to the relation extraction task is the training Amharic free text dataset. Using the train/test splitter, 80 percent of the corpus assigned as training data, while 20% assigned as test data. The training datasets divided into two categories: unlabeled (word embedding dataset) and labeled (word embedding dataset). Which means our model train by the predefined tagged labeled dataset and we test by unlabeled dataset or using free text after the model created. The corpus must go through the primary preprocessing processes before moving on to the following steps. Preprocessing refers to the task of cleansing text data in order to make it usable for analysis. To construct a word vector, not every word in the text is necessary. Text data, for example, frequently comprises particular the most prevalent words, all of which obstruct relation extraction. As a result, text data has preprocessed before use. Cleaning, sentence segmentation, tokenization, and stop word removal are among the proposed system preprocessing operations.

a) Cleaning

The cleaning method removes non-Amharic characters from the collected Amharic corpus in the first stage. The faults found in the quarter of the corpus used to rectify misspelled Amharic terms. This is because the presence of non-Amharic text has influenced the automatic feature generation's performance. As a result, all texts that are not in Amharic eliminated from the corpora.

The following algorithm used for cleaning non-Amharic text from Amharic corpus.

```
Algorithm cleaning ()
        Input: Amharic-corpus

        Output: Cleaned Amharic Text Read List of Amharic characters

           For i in Amharic-corpus do

              If (Amharic corpus[i] == 'non-Amharic character') then

                  Remove Amharic-corpus[i];

              End If

           End for

End algorithm
```

Algorithm 3. 1 Cleaning non-Amharic text

a) Sentence segmentation

The task of breaking a string of written language into its component sentences known as sentence segmentation (Jurish, 2014). This accomplished by identifying sentence boundaries (the ending point of a sentence and the beginning of the next sentences). In Amharic, the techniques used to split the sentences are full stop, interjection, question mark, exclamation mark. Amharic segmenter using python following the algorithm below does this sentence segmentation.

Algorithm sentence-segmentation ()
Input: cleaned Amharic-corpus (CAC)

Output: list of segmented sentences

Sentences=[]

For every character in CAC do

If character is different from punctuation mark ።[!, ?, !!] then

Concatenate character to sentence

Else sentenceList =

sentence

Sentence = []

End if

End for

End algorithm

Algorithm 3. 2 Sentence Segmentation

a) Tokenization

Tokenization is the process of breaking down a given text into tokens, or separating texts into sentences, or sentences into individual words (Jurish, 2014). This accomplished by identifying word boundaries (the ending point of a word and the beginning of the next word). Words, integers, punctuation marks, special symbols, and other symbols can used as tokens. A popular technique to break a text in Amharic, for example, is to use a set of rules as a marker, such as whitespace or punctuation letters. In older Amharic writing styles, ሁለት ነጥብ '፡' used to divide words; presently, white space used instead. We employed different Amharic punctuation signs like ፡, ፣, ፤ and white spaces to identify tokens in this study. It also takes abbreviations and hyphenated words into account. Words like / and -, for example, are treated as a single word. The Amharic tokenizer word splitter is used to tokenize the data, as shown in the below algorithm.

Algorithm word-tokenization ()
Input: Segmented Amharic Sentence (F)

Output: Tokenized Amharic text into list of words
For every word in F do
If punctuation marks or White space found then
Split sentence into words
End if
End for
End algorithm

Algorithm 3. 3 Tokenization

d) Stop word removal

Stop-words are terms that appear frequently in text data but are either irrelevant or have no effect on text discrimination. The most frequently used words in any language are stopwords. When training, they place less pressure on the model. Stopwords are words with minimal meaning that removed. As a result, the model can concentrate on the terms that will be more effective during training.

The Amharic language, like other languages, has a variety of stop-words such as articles, prepositions, and conjunctions. Stop words in Amharic, for example, ስለ, ያለ, እና, ነገርግን, ሁሉም, ኋላ, ሁኔታ, ሆነ, ሆኑ, ሆኖም, ሁሌ, ሁልጊዜ, ሁሉንም, ላይ, ሌላ, ሌሎች, ልዩ, መሆኑ, ማለት, ማለቱ, መካከል etc. non informative using these words in a dataset in their current form will have an effect on the results.

```
Algorithm Stop -word removal
    () input: list-of-tokens, stopword-
    list

Output: stop-words free Amharic text

 For word  in in list-of-tokens do

      For stopword  in stopwords-list do      If

word == stopword THEN        eliminate

word from list-of-tokens

         End if

      End For

   End for

End algorithm
```

Algorithm 3. 4 Stop word removal

e) Trimming

As recommended by the literature, we have condensed each phrase into a shorter piece that only includes the entities, it maybe two, three or four words that go between them and a few words before and after. The trimming operation's goal is to eliminate any sentence components that are irrelevant to the relation extraction.

```
Algorithm trimming ()
      Input: Amharic sentences
      Output: Sentence Components relevant for relation extraction
Window size = 3k first_index = max (tokens. index("ENTITY1") - window_size, 0)

          second_index = min (sentence. index("ENTITY2") + window_size,

          len(tokens)) trimmed_tokens = tokens [first_index: second_index]

end algorithm
```

Algorithm 3. 5 trim

3.3 Feature extraction

Words are fundamental building blocks of letter-formed language. We reproduce noises that make use of a variety of understandable characters. NLP, however the linguistic syntax hierarchy's atomic units of syntax known as morphemes are the only ones that cannot split into smaller components (Dinku, 2020; F. Xu et al., 2019).

After the unlabeled Amharic, free text passes in preprocessing steps each word must maps into a low dimension vector for automatic feature generation by using the word embedding techniques Word2vec (Aschenaki Abi Abera, Yaregal Assabe , Mesfin Kifle , 2020).

Word2vec is a tool for creating a distributed representation of words (Mikolov, 2014). The closer the word meanings are to one another when the tool assigns each word a real-valued vector, the more similarity the vectors imply. In a distributed representation, each word given a real-valued vector then after represented by the vector. We refer to word embedding is when a word is represented by a dispersed representation.

```
Begin
        Input: Amharic Text Corpus
                Tokenize the text ()
                Add all text in one file F ()
                   Call Word Vector Function ()
                              Train F (Word2Vec (F)) ()
                    Save the trained Model
End Algorithm
```

Algorithm 3. 6 Amharic word embedding model

3.4 Pre trained word embedding

The first stage is to create pre-trained word embedding’s, which distributed representations of words acquired from a text corpus as real-valued vectors using neural networks and matrix factoring techniques. These are dense, low-dimensional vector representations of words in a continuous embedding space, where words that semantically linked clustered together and vectors represent similarity. Word embedding is also particularly effective in NLP applications since it maintains both semantic and syntactic terms dependent on their surroundings.

In this study, we employ Continuous Bag of Words, which based on two pioneering efforts using word2vec based on CBOW and skip-gram models. In order to compare pre trained word embedding data with, we are using word2vec file from internet, which contains approximately 304,469 words and vectors.

Model Builder

The main goal of a model builder has to create a trained model using Attention based BiLSTM learning algorithms. For the real training process, all of the previously retrieved features utilized. The training of the model carried out by the model builder. Based on the extracted characteristics, it intended to estimate the model coefficients and then provide a trained model. The sigmoid classifier, which categorizes Amharic sentences containing only two entities or more entities into relation types after extracting its features, trained using the training data created in the previous stage. The model includes a set of numbers that calculated throughout the estimating process. Here, the trained model is a collection of variables (known as weights) that represent the significance of the task-relevant attributes. It represents an estimation of every parameter learned through training. The final output of the learning process, the trained model, serves as an inference for the prediction process.

3.5 Relation extractor

The trained model provides the knowledge needed for the relation extractor to derive relations from the testing data, assisting the relation extractor. The trained model feeds the characteristics that retrieved and saved during training to the relation extractor, which uses them to determine the type of relation from the text.

The most and primary functions of relation extractor are relation detection and relation classification. The initial step in relation extraction is relationship detection, which is the process of finding relationships between mentions of the corresponding entities. The second step in relation extraction is relation classification, which involves classifying the identified relation mentions into a set of predetermined relations. We use a sigmoid classifier to predict a multilabel from a discrete set of multi-classes for a text. By using the trained model, the relation extractor chooses potential relations based on the computed likelihood.

Finally, the outputs of Amharic predicted relations displayed as follows but the determinant is the sentence contained the number of entities because a sentence may contain only two entity or three and maximum four entity at this time the predefined relation type integrate different position indictors.

የተመረቁበት_ትምህርት_ቤት (e1, e2), የተወለዱበት_ዘመን (e1, e2), ትምህርታቸውን_በመከታተል_ላይ (e1, e2), የስራ_ቦታ (e1, e2) መስራች (e1, e2), የተወለዱበት_ቦታ (e1, e2) ባለቤት (e1, e2), አባል (e1, e2)የለም (e1, e2), መሪ (e1, e2) መዳረሻ (e1, e2), ዜግነት (e1, e2) ሙያ (e1, e2), ዋና_ከተማ (e1, e2) ሀላፊ (e1, e2), አካባቢ_ውስጥ_ይገኛል (e1, e2), የሞቱበት_ዘመን (e1, e2), አምራች (e1, e2) ጸሀፊው_ነው (e1, e2), ትምህርታቸውን_ተከታትለዋል, (e1, e2)የሞቱበት_ቦታ (e1, e2), ሌሎች (e1, e2) ነዋሪ (e1, e2), ተዋሰነ (e1, e2), አካል (e1, e2), መነሻ (e1, e2)

CHAPTER FOUR፥

EXPERIMENTATION AND DISCUSSION

4.1 Overview

This chapter discusses the experimental results by showing experimental setups and performance of testing results of the systems using accuracy score metrics. In addition, we have compared the results of the experimental systems between the language pair in both directions. In this chapter, we have followed the following steps. The results obtained from the models then subjected to the evaluation metrics. Using various evaluation metrics can represent a better interpretation of the evaluation of the results.

The laptop machine used for conducting experiment this study has a memory capacity of 6 GB RAM, 2.2 GHz processor, and 64-bit Operating system, x64 based processor for training the task of relation extraction using deep learning algorithm and achieving good result from training.

4.2 Data collection and preparation

In our knowledge, there is no public dataset to conduct our experiments. Therefore, we prepare Amharic semantic relation extraction dataset called (Amharic_SRE_dataset) to conduct the experiments for this research. The corpus annotated with five entity types i.e. person, organization, time, books and product, with 26 predefined relation types. The preparation shares the concept of SemEval2010 Task8 public dataset.

The Amharic_SRE_dataset contains 26 direct relation labels; these are መስራች, ባለቤት, ሀላፊ, የስራ_ቦታ, የትውልድ_ቦታ, የትወልድ_ዘመን, ነዋሪ, አባል, መነሻ, መዳረሻ, ዜግነት, አካል, ጸሀፊው_ነው, ዋና_ከተማ, አካባቢ_ውስጥ_ይገኛል, ትምህርታቸውን_በመከታተል_ላይ, ተዋሰነ, የተመረቁበት_ትምህርት_ቤት, ሙያ, የሞቱበት_ዘመን, የሞቱበት_ቦታ, ሌሎች and የለም. All these relation types explained between two-named entities. This named entity called based on noun-to-noun examples as follows.

- መስራች this type of relation exists between PER and ORG which means that a person founds the organization as the first time. The organization founded by a person. For example: <e1>ኢሎን ማስክ</e1> <e2>የቴስላ ሞተርን</e2> መሰረተ።

- ባለቤት this type of relation exists between PER and ORG which means a person owns an organization. For example: <e1>ዙከምበርግ</e1> <e2>የፌስቡክ</e2> ባለቤት ነው ።

- ሀላፊ A relation falls in between PER and ORG which a person leads the organization and expects high decision, managing the company resources. For example: አቶ <e1>ምሕረት ደበበ</e1> ፣<e2>የኢትዮጵያ ኤሌክትሪክ ኃይል ኮርፖሬሽን</e2> ዋና ሥራ አስፈፃሚ <e3>የግቤ ሁለት ኃይል ማመንጫ</e3> ሥራ ጀምሯል ።

- የስራ_ቦታ a relation falls in between PER with Either LOC or ORG and works on specified location or an organization. For example: <e1>አለሙ</e1> <e2>በባህር ዳር ዩኒቨርሲቲ</e2> መምህር ነው ።

- የትውልድ_ቦታ a person birthplace. For example: <e1>ከበደ</e1> <e2>በባህር ዳር ከተማ</e2> ተወለደ።

- የትወልድ_ዘመን a person date of birth. For example: <e1>ከበደ</e1> <e2>በ1996 ዓ/ም </e2> ተወለደ።

- አባል A PER or a LOC is a member of a group, political party, football team and so on. For example: <e1>የህወሓት</e1> ጁንታ አባላት አንዷ የሆኑት ወይዘሮ <e2>ኬሪያ ኢብራሂም</e2> <e3>ለመንግስት</e3> እጅ ሰጡ ።

- ዋና_ከተማ the city or town that functions as the seat of government and administrative centre of a country or region. For example: <e1>የትግራይ ክልላዊ መንግሥት</e1> ዋና መዲና በሆነችው <e2>በመቀሌ ከተማ</e2> ተወዳጁ ድምፃዊ <e3>ዳዊት ነጋ</e3> ተወለደ ።

Finally, we have collected 25,000 sentences of Amharic corpus. The corpus that has gathered are from different social media platforms.

Source of data	**No of sentences**
Facebook from governmental pages	200
Wikipedia	**1800**
Personal blog	200
Broadcast media	100
University posts	100

Total	2,500

Table 4. 1source and size of data for experiment

These 2,500 sentences divided into a training data set and a testing data set, which we employed for our study. Before classifying the dataset, we have shuffled the data. After shuffling the data, we classified the dataset in to training and testing. We have used 80 percent of the dataset for training and 20 percent of the dataset for testing the relation classification model. The reason that we used 80/20 is because, most of the related works classify their data by using pareto principle (80/20) techniques, which states that 80 percent of the total dataset is used for training and 20 percent is left for testing the system. As a result, the training dataset contains 2,000 sentences and the test dataset contains 500 sentences.

4.3 Experimental setups

After we finished preparing the dataset, we used a paper space with 30 GB of RAM and a GPU RTX5000, which used to process with a short training time. To build our system, we used the Python programming language along with the Keras, Tensorflow, NumPy, and PyTorch libraries. We conducted the experiments using word and sentence as the relation extraction unit using different classical machine learning algorithm and LSTM and BiLSTM.

As discussed above our dataset have 26 predefined semantic relation types i.e., የተመረቁበት_ትምህርት_ቤት, የተወለዱበት_ዘመን, ትምህርታቸውን_በመከታተል_ላይ, የስራ_ቦታ, መስራች, የተወለዱበት_ቦታ, ባለቤት, አባል, የለም, መሪ, መዳረሻ, ዜግነት, ሙያ, ዋና_ከተማ, ሀላፊ, አካባቢ_ውስጥ_ይገኛል, የሞቱበት_ዘመን, አምራች, ጸሀፊው_ነው, ትምህርታቸውን_ተከታትለዋል, የሞቱበት_ቦታ, ሌሎች, ነዋሪ, ተዋሰነ, አካል or መነሻ. Hence, our problem can belong to multiclass or multi-label classification problem because we can recognize the named entity by using named entity recognizer. As we can see the above description this problem is, a multilabel classification but we also conduct multiclass classification problem.

Multilabel Classification: This type of classification involves examples such that each example can falls to multiple categories and not necessarily only one category. A multi-label classification problem viewed as a generalized version of the multiclass classification problem in which there is no restriction over how many classes a training example can belong to (Dinku, 2020)(Kumar, 2021)(Aggarwal & Tiwari, 2021).

As in our dataset, our data can belong to one to any number of categories; so, our problem is Multilabel Classification problem. Even if we also conduct the multi-class classification problem because of extracting the semantic relation between named entities have more relation types. We tried the experiments then have get encourage able results.

Multiclass Classification: In this classification, we put each example into one of the several possible categories unlike binary classification problem where each example belongs to only one of two possible categories (Aggarwal & Tiwari, 2021; Dinku, 2020; Kumar, 2021).

4.4 Parameter selection

In our experimental study, we trained and tested the model using our sentence-based corpus as prepared. To achieve the desired result, we conducted various experiments on various parameters using our training data. We started by selecting embedding dimension. We selected the embedding dimension 128 and 256. Next, we selected batch size, 4, 8, 16 based on the related works. In order to select the best dimension using the training data we have done the experiments using both embedding dimension with each batch size. Here to select the best, we have used 0.2recurrent dropout rate, Adam optimizers, and dropout rate, which is 0.2. Then we have done the experiments with 5 epochs. Finally, we have got the following results specified in the Table 4.2. These results are almost the same, only slightly different. In all experiments below, the loss level are goes in the same way from higher loss level to lower loss level.

Embedding dimension	**Batch size**	**Training loss**	**Time taken(seconds)**
128	4	0.1508	84
128	8	0.0826	41
128	**16**	**0.0597**	**21**
256	4	0.0647	84
256	8	0.1386	43
256	16	0.2297	21

Table 4. 2Training time and Loss for selecting embedding and batch size

As we have seen the above experiment results, the best that we have, are with 128 embedding dimension and 16-batch size. Therefore, we have chosen embedding dimension 128 and batch size 16. The time taken for embedding dimension 256 is the same. They only differ in microseconds.

The next step is choosing dropout rate. We have compared the above best experiments with 0.2, 0.5, and 0.7-dropout rate with recurrent dropout rate 0.1 and batch size 16 using 5 epochs based on the related works. In addition, as we have seen in the Table 4.3, we have minimal loss with dropout rate 0.2.

Embedding dimension	Recurrent dropout rate	Dropout rate	Training loss
256	**0.2**	**0.2**	**0.1378**
256	0.2	0.3	0.1728
256	0.2	0.5	0.1775
256	0.2	0.7	0.1785

Table 4. 3Training Loss for selecting Dropout rate

Finally, we selected the recurrent dropout rate. We started with large value like, 0.1 and then, we tried with values: 0.2, 0.3 respectively, and its loss is 0.1531, 0.1508, and 0.1509 respectively. Finally, we got best result with learning rate 0.2.

Hyperparameters are characteristics of training data that the classifier or other deep learning models will train on their own. They include parameters that control the network's structure (such as the number of hidden units) and its training process (e.g: learning rate). The performance of the model trained significantly influenced by the behavior of the training algorithm, which directly controlled by hyperparameters (Journal & Technology, 2019).

Given the effect on the learnt model, selecting the right hyperparameters is essential to the success of neural network architectures. The benefits of selecting effective hyperparameters include an effective search throughout the space of potential hyperparameters and the ability to manage the experiment (Aghaebrahimian & Cieliebak, 2019). For example, the following are some model inbuilt configuration variables:

For training, we considering Amharic free text semantic relation classification dataset at sentences level with an input size of MAX_NB_WORDS = 30000. A dropout rate of 0.2 used to regularize the network parameters with training epoch typical value 5.

4.5 Experimental results

In our experiment, we conducted Amharic semantic relation classification from free texts Using Logistic Regression, LSTM, Bi-LSTM, CNN-BiLSTM and Transformer. We have followed two scenarios to conduct our experiments. These are multi-class single output and multi-label classification. Experimental results presented in sub section 4.5.1 and 4.5.2.

4.5.1 Multiclass classification experiments

In Table 4.4 below, we presented the experimental result for multiclass relation classification.

Model	Average	F1-score	Precision	Recall
LSTM	Makro	0.53	0.57	0.57
	Weighted	0.58	0.62	0.60
Bi-LSTM	Makro	0.55	0.61	0.62
	Weighted	0.59	0.64	0.60

Table 4. 4 multiclass relation classification experimental results

The first experiments that we conducted here is using LogisticRegression classical machine learning algorithm with tfidf word embedding techniques. Here we have seen the training time and accuracy score metrics to measure the performance using 20% of testing data with random state 7. For training our relation classification using Logistic Regression, the model has taken 32 seconds Amharic relation extraction in single label output respectively. The accuracy score that we have 0.73.

The second experiments that we conducted here is using LSTM. Here we have seen the training time and accuracy score metrics to measure the performance using 5 epochs. For training our relation extraction using LSTM, the model has taken 62 seconds Amharic relation extraction in single label output respectively. The accuracy score that we have 0.49.

The third experiments are using Bi-directional LSTM. Bi-LSTM has double LSTM cells in encoder side, which take a lot of memory than LSTM. Here we have used the same parameters as we used during LSTM. In addition, Bi-LSTM model has taken 124 seconds to extract Amharic relation in single output. In addition, we have 0.62 accuracy score.

Sample multiclass relation classification result shown as follows in table 4.5

Input sentence	**Predicted relation**
የአፍሪካ አንድነት ድርጅት ቢሮው በኢትዮጵያ መዲና አዲስ አበባ ከተማ	ዋና_ከተማ
የአፍሪካ አንድነት ድርጅት ቢሮው በኢትዮጵያ መዲና አዲስ አበባ ከተማ	መዳረሻ
ክቡር ዶ ር አርቲስት ጥላሁን ገሰሰ በተወለዱ በ68 ዓመቱ ሚያዚያ 11 2001 ዓ/ም በአዲስ አበባ አለም በሞት ተለዩ	የሞቱበት_ዘመን
ክቡር ዶ ር አርቲስት ጥላሁን ገሰሰ በተወለዱ በ68 ዓመቱ ሚያዚያ 11 2001 ዓ/ም በአዲስ አበባ አለም በሞት ተለዩ	የሞቱበት_ቦታ

Table 4. 5 sample multiclass relation classification

4.5.2 Multi-label relation classification experiment’s

Model	**Accuracy**
Logistic regression	47%
LSTM	39%
BiLSTM	55%
CNN-BiLSTM	52%

Table 4. 6 the experimental results for multi-label relation classification.

The first experiments that we conducted here is using Logistic Regression classical machine learning algorithm with TFIDF word embedding techniques. Here we have seen the training time and accuracy score metrics to measure the performance using 20% of testing data with random state7. For training our relation extraction using Logistic Regression, the model has taken 62 seconds Amharic relation extraction in single label output respectively. The accuracy score that we have 0.47.

The second experiments that we conducted here is using LSTM. Here we have seen the training time and accuracy score metrics to measure the performance using 50 epochs. For training our relation extraction using LSTM, the model has taken 62 seconds Amharic relation extraction in multi label output respectively. In addition, we have 0.39 accuracy score.

The third experiments are using Bi-directional LSTM. Bi-LSTM has double LSTM cells in encoder side, which take a lot of memory than LSTM. Here we have used the same parameters as we used during LSTM. In addition, Bi-LSTM model has taken 119 seconds to extract Amharic relation in multi label output. In addition, we have 0.55 accuracy score.

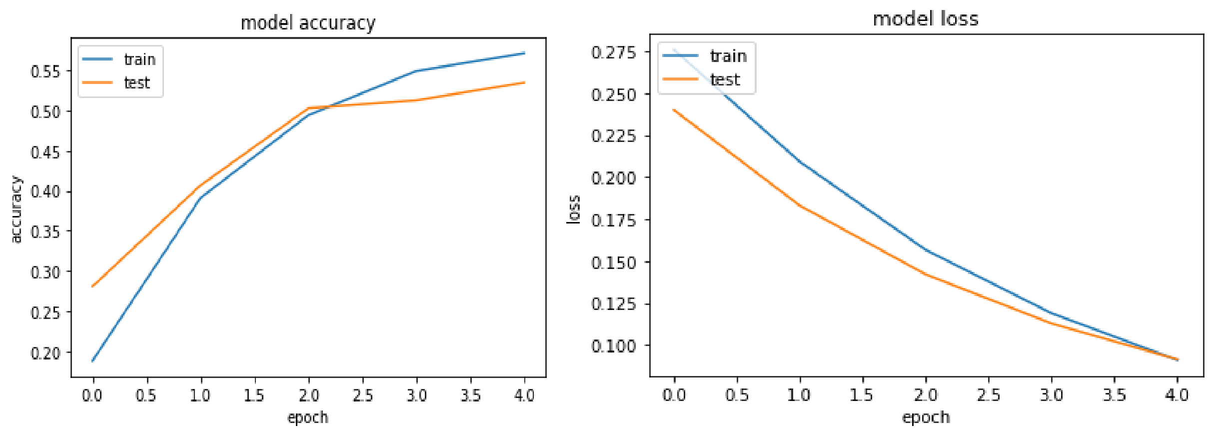

Figure 4. 1 the training and validation loss of the BiLSTM model

The fourth experiments are using CNN-BiLSTM. The convolutional layer extracts the feature in the first layer then LSTM layer as seq2seq learner to get the final desired output. Here we have used the same parameters we used during LSTM and plus spatial dropout 0.2, conv1D 64 and kernel size is 3.the model has taking 230 seconds to extract a relation between Amharic named entities. The model returns the test accuracy with unseen data is 0.52 accuracy score.

The model looks like figure 4.1 in graphical representation

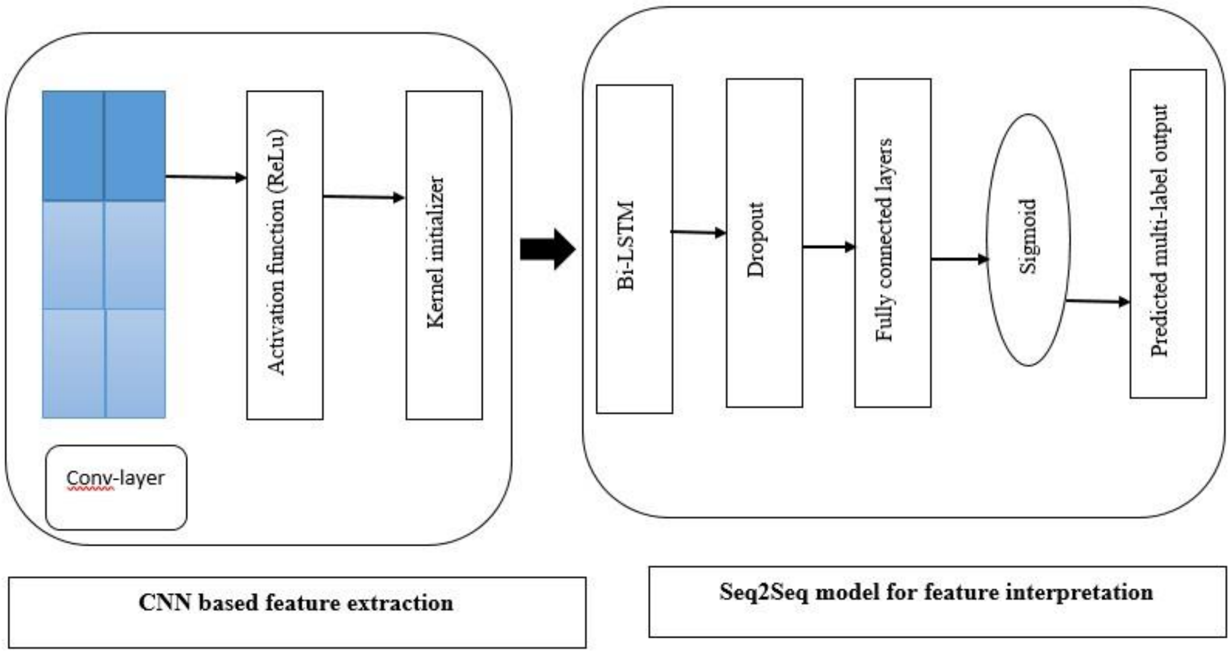

Figure 4. 2 multi-label text classification using CNN-BiLSTM

Sample multi-label relation classification as follow

Input sentence	Predicted first relation one	Predicted second relation
አቶ በለጠ በባህር ዳር ዩኒቨርሲቲ ተመርቆ መምህር	የተመረቁበት_ትምህርት_ቤት	የስራ_ቦታ
ወ/ሮ መዓዛ የሮሃ ሚድያ መስራችና ባልቤት	መስራች	ባለቤት
አለሙ በባህር ዳር ከተማ በ1980 ዓ/ም ተወለደ	የተወለዱበት_ዘመን	የተወለዱበት_ቦታ
አቶ አቤል በአዲስ አበባ ዩኒቨርሲቲ መምህር እና የሁለትኛ ዲግሪ ተማሪ	የስራ_ቦታ	ትምህርታቸውን_በመከታተል_ላይ

Table 4. 7 multi-label relation classification

4.6 Discussion result

The main purpose of this study is to conduct experiment on relation extraction between named entities from Amharic free text using different machine and deep learning approaches. Different experiments conducted using multi-class single label output and multi-class multi output. To achieve the goal of this thesis work, we concentrated on the design and implementation of entity relation classification from Amharic free texts. In addition, we have best accuracy score result in single label output. However, we have implemented both multi-class single output and multiclass multi-output using five different algorithms.

In addition, we suggested multi-label relation classification using BiLSTM models. Because as we have shown the experimental results, multi-label relation classification, using BiLSTM models are effective with minimal time, best accuracy score result comparing with others. Even when we compared with memory, BiLSTM model is best, effective a memory usage. The main challenges that we have faced during LTSM and Bi-LSTM is, they take a lot of memory. We were unable to execute our LSTM and Bi-LSTM models in our best parameters using 12 GB RAM. Due to that we upgraded the memory to 32 GB RAM. The advantage of BiLSTM model is, for high length sentence, the interdependence of words at the beginning and end of the sentence becomes more comparing with others. This is because context within a sentence derived as the interdependence of all previous words in a given sequence of words in the sentence using the forward and backward propagation cells. However, it is more context dependent; it works well for long sentences.

Our experiments conducted the above hyperparameters in addition we tried by using two word embedding system. This is BiLSTM default Keras embedding layer and word2vec embedding techniques during train the model. Word2vec as a word embedding techniques we use and embed around 40,000 sentences from our corpus. Therefore, the word2vec returns highly encourage able results however; the Keras a little bit encouraged result. From this, we conclude that a word2vec embedding technique has selected for this thesis.

Therefore, our selected model is multi-label relation classification using BiLSTM models. Here bellows a comparison also made with various previous studies. Most of our local language previous studies conducted using multi-class single output using deep learning approaches. However, only one multi-class single output relation classification studied using deep learning approaches in our local languages (Aschenaki Abi Abera, Yaregal Assabe, Mesfin Kifle, 2020). Aschenaki Abi studies have nine predefined relation types namely Entity-Origin, Entity-Destination, Product-Producer, Member-Collection, Message-Topic, Content-Container, Cause-Effect, Component-Whole and other. According to the researcher's knowledge, there is no research studied using BiLSTM model in our local languages for relation classification multilabel. As to the researcher's knowledge, there is no previous study conducted triple and multiple relation classification between Amharic languages named entity pairs. However, our study is conducted Amharic triple and multiple relation classification. We have used the classical machine-learning algorithm and the two public RNN algorithms known as LSTM and Bi-LSTM model. Experiments that we conducted using these models are in both multi-class single output and multi-label relation classification. To conduct our experiment, we have collected the dataset since there was no available standardized prepared corpus on Amharic language entity pairs prepared to this purpose.

Generally, we have answered the first research question during the preparation of the dataset using entity indicators. Entity indicators carry the information of named entities' starting and ending positions. Then a sentence having only two entities can have a relation either single or triple relation but not have multiple relations whereas a sentence with more than two entities has multiple relations but not triple relation.

CHAPTER FIVE፥
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

There are many electronic documents available today on the Internet and on mobile devices that can aid people with their day-to-day lives. Finding effective methods to automatically index and analyze texts is required due to the growing amount of content that is available online. This volume of data is too large for humans to process, and searching does not lessen it. Therefore, it is necessary to synthesize the available content and extract the key information. However, the abundance of information makes it challenging to manually sift through the sea of unstructured data and gather the necessary information. The amount of text data in other languages is likewiseprogressively growing. This is also true for Amharic, since there has been an increase in the creation and consumption of various online publications and contents.

In this research work, we have developed relation extraction for Amharic language free text. The system has three basic components developed using BiLSTM. The first component is the preprocessing module, which resolves language specific issues and makes the data ready for extraction. The second one is the main unit of the extractor system, which called relationextraction module. The extractor component used to identify entity categories and select the relevant one. Then the extracted entities presented using annotations. When testing our system, the system evaluation shows a promising performance. We have used 2,500 sentences fortraining, testing, and obtained 55% F1-measure. In general, given different constraints ouralgorithm obtained good performance compared with resource rich languages like English.Finally, the scarce of the data plus the appropriate algorithm selection and the unseen more semantic relation exists between the named entities. In addition, our works concerned only with the five entities. The relation extraction has concerned only the noun to noun however; we can also extract the relation resolving Coreference.

5.2 Contribution

The main contributions of the study outlined below:

- Prepared new suitable Amharic semantic relation extraction dataset
- The general architecture for relation extractor from Amharic language free text

- Algorithms developed for language specific issues, which handle normalization and tokenization.
- Develop the semantic relation classification model
- Implement Amharic language free text relation extractor
- Conduct experiment and come up with promising result.

5.3 **Recommendation**

Relation extraction is probably a new study for Amharic language. The task is very complex for such under resourced languages. The developed relation extraction system for Amharic language has portions that require further improvements that we want to recommend them as future works. The following are our recommendations:

- The size of the training and test collection used in this research is too small because of the scarce of the However;one can increase the data collection and improve the model performance.
- In this work, only directly named entity considered. Therefore, co-referencing will increase the performance of extraction and are highly recommended.
- For recognizing names, the rules, which depend on, sentence pattern used. However, sentence cannot always be in the same pattern. Therefore, using an automatic named entity recognition in later stages might minimize the burden of selecting the named entities
- Incorporating Amharic spell checkers to minimize the spelling problems, which mostly happen in the news text, might also have an impact as we manually modify the spelling errors as they have impact for named entity recognition.
- Incorporating Amharic word net for understanding the sense of words so that extraction will be better

References

"Deep-learning/what-is-the-difference-between-sigmoid-and-softmax-activationfunction/," 20 April 2020. [Online]. Available: https://nomidl.com/nomidl.com/deeplearning/what-is-the-difference-between-sigmoid-and-softmax-activation-function/.

Addis Ababa, Ethiopia March 2020. (2020). *March*.

Adnan, K., & Akbar, R. (2019). An analytical study of information extraction from unstructured and multidimensional big data. In *Journal of Big Data*. Springer International Publishing. https://doi.org/10.1186/s40537-019-0254-8

Aggarwal, A., & Tiwari, A. (2021). *Multi Label Toxic Comment Classification using Machine Learning Algorithms*. *3878*(1), 158–161. https://doi.org/10.35940/ijrte.A5814.0510121

Aghaebrahimian, A., & Cieliebak, M. (2019). *Hyperparameters Tuning for Deep Learning in Natural Language Processing*.

Agosti, M., Di Nunzio, G. M., Marchesin, S., & Silvello, G. (2019). A relation extraction approach for clinical decision support. *CEUR Workshop Proceedings*, *2482*(May).

Al Zamil, M. G. H., & Al-Radaideh, Q. (2014). Automatic extraction of ontological relations from Arabic text. *Journal of King Saud University - Computer and Information Sciences*, *26*(4), 462–472. https://doi.org/10.1016/j.jksuci.2014.06.007

AlArfaj, A. (2019). Towards relation extraction from Arabic text: a review. *International Robotics & Automation Journal*, *5*(5), 212–215. https://doi.org/10.15406/iratj.2019.05.00195

Alom, Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Essen, B. C. Van, Awwal, A. A. S., & Asari, V. K. (2019). *A State-of-the-Art Survey on Deep Learning Theory and Architectures*. 1–67. https://doi.org/10.3390/electronics8030292

Alotayq, A. (2013). Extracting relations between Arabic named entities. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8082 LNAI*, 265–271. https://doi.org/10.1007/978-3642-40585-3_34

Appelt, D. E. (1999). Introduction to Information Extraction. *AI Communications*, *12*(3), 161–172.

Aschenaki Abi Abera, Yaregal Assabe (PhD), Mesfin Kifle (PhD), A. B. (PhD). (2020). No Title. *Semantic Relation Extraction for Amharic Text Using Deep Learning Approach Aschenaki Abi Abera A*.

Ashagrie, F., & Boran, D. (2019). Ancient Geez script recognition using deep learning. *SN Applied Sciences*, *1*(11), 1–7. https://doi.org/10.1007/s42452-019-1340-4

Asker, L., Alemu, A., & Bj, A. (n.d.). *Classifying Amharic Webnews*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A nucleus for a Web of open data. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4825 LNCS*, 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

Baader, F., Horrocks, I., & Sattler, U. (2004). Description Logics. In *Handbook on Ontologies*. https://doi.org/10.1007/978-3-540-24750-0_1

Benuwa, B., Zhan, Y., & Ghansah, B. (2016). *A Review of Deep Machine Learning*. *February 2017*. https://doi.org/10.4028/www.scientific.net/JERA.24.124

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (1997). Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, *26*(2), 1247–1249.

Boudabous, M. M., & Chaâben, N. (2013). *Arabic WordNet semantic relations enrichment through morpho-lexical patterns*.

Boujelben, I., Jamoussi, S., & Ben Hamadou, A. (2014). A hybrid method for extracting relations between Arabic named entities. *Journal of King Saud University - Computer and Information Sciences*, *26*(4), 425–440. https://doi.org/10.1016/j.jksuci.2014.06.004

Brambilla, M., Celino, I., Ceri, S., Cerizza, D., Valle, E. Della, & Facca, F. M. (2006). The Semantic Web - ISWC 2006. *International Semantic Web Conference*, *4273*(May 2014), 172–186. https://doi.org/10.1007/11926078

Burget, L. (2010). *Recurrent neural network based language model ´ s Mikolov*. *May 2014*.

Cascade-correlation, R., & Chunking, N. S. (1997). *2 PREVIOUS WORK*. *9*(8), 1–32.

Dai, Z., Li, L., & Xu, W. (2016). CFO: Conditional Focused neural question answering with large-scale knowledge bases. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, *2*(1), 800–810. https://doi.org/10.18653/v1/p16-1076

Date, E. (1995). Ethiopia. *Ethiopian Constitution*.

Demeester, T., Rocktäschel, T., & Riedel, S. (2016). Lifted rule injection for relation embeddings. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 1389–1399. https://doi.org/10.18653/v1/d16-1146

Devisree, V., & Raj, P. C. R. (2016). A Hybrid Approach to Relationship Extraction from Stories. *Procedia Technology*, *24*, 1499–1506. https://doi.org/10.1016/j.protcy.2016.05.101

Dinku, W. (2020). *Multi Label Amharic Text Classification Using Convolutional Neural Network Approaches*.

Feng, Y., Zhang, H., Hao, W., & Chen, G. (2017). Joint extraction of entities and relations using reinforcement learning and deep learning. *Computational Intelligence and Neuroscience*, *2017*. https://doi.org/10.1155/2017/7643065

Goundar, S., & Universities, M. (2019). *Chapter 3 – Research Methodology and Research Method*. *May*.

Graves, A. (n.d.). *Generating Sequences With Recurrent Neural Networks*. 1–43.

Grishman, R. (2015). *Relation Extraction : Perspective from Convolutional Neural Networks Relation Extraction : Perspective from Convolutional Neural Networks*. *June 2016*. https://doi.org/10.3115/v1/W15-1506

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (n.d.). *Improving neural networks by preventing co-adaptation of feature detectors arXiv : 1207 . 0580v1 [cs . NE] 3 Jul 2012*. 1–18.

Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://doi.org/10.1007/s10994-019-05855-6

Huang, Z., Xu, W., & Yu, K. (2015). *Bidirectional LSTM-CRF Models for Sequence Tagging*. http://arxiv.org/abs/1508.01991

Huminski, A., & Bin, N. Y. (2020). Automatic extraction of causal chains from text.

Libres, *29*(2), 99–108. https://doi.org/10.32655/libres.2020.29.2.3

Igrejas, G., Fdez-, F., & Martín, P. (2022). *A deep learning relation extraction approach to support a biomedical semi-automatic curation task : The case of the gluten bibliome*. *195*. https://doi.org/10.1016/j.eswa.2022.116616

Journal, I., & Technology, R. (2019). *Hyperparameter Optimization and Regularization on Fashion-MNIST Classification*. *2*, 3713–3719. https://doi.org/10.35940/ijrte.B3092.078219

Kumar, A. (2021). *Multilabel Toxic Comment Detection and Classification*. *10*(05),

55–61.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 260–270. https://doi.org/10.18653/v1/n16-1030

Lee, J. (2010). *Semantic Relation Classification via Bidirectional LSTM Networks with Entity-aware Attention using Latent Entity Typing*. *1*.

Lee, J. Y., Dernoncourt, F., & Szolovits, P. (2018). *MIT at SemEval-2017 Task 10: Relation Extraction with Convolutional Neural Networks*. 978–984. https://doi.org/10.18653/v1/s17-2171

Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M., & Li, J. (2020). Entityrelation extraction as multi-turn question answering. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 1340–1350. https://doi.org/10.18653/v1/p19-1129

Liu, K., Chen, Y., Liu, J., Zuo, X., & Zhao, J. (2021). Extracting Events and Their

Relations from Texts : A Survey on Recent Research Progress and Challenges. *AI*

Open, *1*(March), 22–39. https://doi.org/10.1016/j.aiopen.2021.02.004

Lv, C., Pan, D., Li, Y., Li, J., & Wang, Z. (2021). A Novel Chinese Entity Relationship Extraction Method Based on the Bidirectional Maximum Entropy Markov Model. *Complexity*, *2021*. https://doi.org/10.1155/2021/6610965

Lv, X., Guan, Y., Yang, J., & Wu, J. (2016). Clinical Relation Extraction with Deep Learning. *International Journal of Hybrid Information Technology*, *9*(7), 237–248. https://doi.org/10.14257/ijhit.2016.9.7.22

Mahendran, D. (2022). *VCU Scholars Compass*.

Martin, D., & Powers, W. (2015). *Evaluation : From precision , recall and F-measure to ROC , informedness , markedness & correlation EVALUATION : FROM PRECISION , RECALL AND F-MEASURE TO ROC , INFORMEDNESS , MARKEDNESS & CORRELATION*. *January 2011*. https://doi.org/10.9735/2229-3981

Mikolov, T. (n.d.). *Exploiting Similarities among Languages for Machine Translation*.

Mikolov, T. (2014). *Efficient Estimation of Word Representations in Vector Space*. *January 2013*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (n.d.). *Distributed Representations of Words and Phrases and their Compositionality arXiv : 1310 . 4546v1 [cs . CL] 16 Oct 2013*. 1–9.

Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using LSTMs on sequences and tree structures. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, *2*, 1105–1116. https://doi.org/10.18653/v1/p16-1105

Muzaffar, A. W., Azam, F., & Qamar, U. (2015). A Relation Extraction Framework for Biomedical Text Using Hybrid Feature Set. *Computational and Mathematical Methods in Medicine*, *2015*. https://doi.org/10.1155/2015/910423

Naeem, M., Tahir, S., Rizvi, H., & Coronato, A. (2020). *A Gentle Introduction to Reinforcement Learning and Its Application in Different Fields*. *8*. https://doi.org/10.1109/ACCESS.2020.3038605

Nandanwar, A. K. (2021). *SS symmetry Semantic Features with Contextual Knowledge-Based Web Page Categorization Using the GloVe Model and Stacked BiLSTM*.

Omar, M., & Abdulla, A. (2021). *The Entities Extraction for Entity Relationship Models from Natural Language Text via Machine Learning Algorithms The Entities*

Extraction for Entity Relationship Models from Natural Language Text via Machine Learning Algorithms. *December 2020*.

Palmer, D. D. (n.d.). *Chapter 2 : Tokenisation and Sentence Segmentation*. 1–23.

Peng, G., & Chen, X. (2020). Entity–relation extraction—a novel and lightweight method based on a gate linear mechanism. *Electronics (Switzerland)*, *9*(10), 1–15. https://doi.org/10.3390/electronics9101637

Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe : Global Vectors for Word Representation*. 1532–1543.

Perera, N., Dehmer, M., Emmert-streib, F., & Emmert-streib, F. (2020). *Named Entity Recognition and Relation Detection for Biomedical Information Extraction*. *8*(August). https://doi.org/10.3389/fcell.2020.00673

Prasojo, R. E. (2016). Entity-relationship extraction from wikipedia unstructured text. *CEUR Workshop Proceedings*, *1733*, 74–81.

Qin, Y., Yang, W., Wang, K., Huang, R., Tian, F., Ao, S., & Chen, Y. (2021). Entity relation extraction based on entity indicators. *Symmetry*, *13*(4), 1–14. https://doi.org/10.3390/sym13040539

Rengasamy, D., Jafari, M., Rothwell, B., & Chen, X. (2020). *Deep Learning with Dynamically Weighted Loss*. *January*. https://doi.org/10.3390/s20030723

Rindflesch, T. C., Tanabe, L., Weinstein, J. N., & Hunter, L. (2000). EDGAR: extraction of drugs, genes and relations from the biomedical literature. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *February*, 517–528. https://doi.org/10.1142/9789814447331_0049

Rybalkin, V., Sudarshan, C., Weis, C., Lappas, J., Wehn, N., & Cheng, L. (2021). *Efficient Hardware Architectures for 1D- and MD-LSTM Networks*.

Salawu, A., & Aseres, A. (2015). *Communicatio : South African Journal for Communication Theory and Research Language policy , ideologies , power and the Ethiopian media*. *April*. https://doi.org/10.1080/02500167.2015.1018288

Sikdar, U. K., & Gambäck, B. (2018). Named entity recognition for amharic using stack-based deep learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in*

Artificial Intelligence and Lecture Notes in Bioinformatics), *10761 LNCS*, 276–287. https://doi.org/10.1007/978-3-319-77113-7_22

Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2016). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*. https://doi.org/10.1016/j.jbusres.2016.08.001

SungMin, Y., SoYeop, Y., & OkRan, J. (2020). DeNERT-KG : Named Entity and Relation Extraction Model Using DQN, Knowledge Graph , and BERT. *Applied Sciences (Switzerland)*.

Taghizadeh, N., Faili, H., & Maleki, J. (2018). Cross-Language Learning for Arabic Relation Extraction. *Procedia Computer Science*, *142*, 190–197. https://doi.org/10.1016/j.procs.2018.10.475

Tan, Z., Shen, Y., Hu, X., Zhang, W., Cheng, X., Lu, W., & Zhuang, Y. (2017). *for Relational Triple Extraction*.

Thankumar, S. I., Velliangiri, S., Alagumuthukrishnan, S., & Thankumar, S. I. (2019). ScienceDirect ScienceDirect ScienceDirect ScienceDirect A Review of Dimensionality Reduction Techniques for Efficient Computation A Review of Dimensionality Reduction b Techniques for Efficient Computation A Review of Dimensionality Reduction Techniques for Efficient. *Procedia Computer Science*, *165*, 104–111. https://doi.org/10.1016/j.procs.2020.01.079

Tietz, T., & Sack, H. (2019). Linked Data Supported Content Analysis for Sociology. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11702 LNCS*, 34–49. https://doi.org/10.1007/978-3-030-33220-4_3

Trisedya, B. D., Qi, J., Zhang, R., & Wang, W. (2018). GTR-LSTM: A triple encoder for sentence generation from RDF data. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 1627–1637. https://doi.org/10.18653/v1/p18-1151

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). *Deep Learning for Computer Vision : A Brief Review*. *2018*.

Worku, B. (2015). *COLLEGE OF NATURAL SCIENCES DEPARTMENT OF COMPUTER SCIENCE Information Extraction from Amharic language Text : Knowledge-poor Approach COLLEGE OF NATURAL SCIENCES DEPARTMENT OF COMPUTER SCIENCE Information Extraction from Amharic language Text : Knowledge*.

Wu, A., Song, Y., Van Oosterom, E. J., & Hammer, G. L. (2016). Connecting biochemical photosynthesis models with crop models to support crop improvement. *Frontiers in Plant Science*, *7*(OCTOBER2016), 305–339. https://doi.org/10.3389/fpls.2016.01518

Xu, F., Zhang, A., & Han, J. (2019). *Parsimonious Morpheme Segmentation with an Application to Enriching Word Embeddings*.

Xu, X., Gao, T., Wang, Y., & Xuan, X. (2022). Event temporal relation extraction with attention mechanism and graph neural network. *Tsinghua Science and Technology*, *27*(1), 79–90. https://doi.org/10.26599/TST.2020.9010063

Yadav, S., Ramesh, S., Saha, S., & Ekbal, A. (2020). Relation Extraction from Biomedical and Clinical Text: Unified Multitask Learning Framework. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–1. https://doi.org/10.1109/tcbb.2020.3020016

Zhang, D., & Wang, D. (2015). *Relation Classification via Recurrent Neural Network.*

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016). Collaborative knowledge base embedding for recommender systems. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 353–362. https://doi.org/10.1145/2939672.2939673

Zhong, Z., & Chen, D. (2021). *A Frustratingly Easy Approach for Entity and Relation Extraction*. 50–61. https://doi.org/10.18653/v1/2021.naacl-main.5

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). *Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification*. 207– 212.

Appendix

Word2vec our dataset

```
[33]
texts= []
for i in dt['texts']:
  texts.append(i.split())
print(texts[:1])
```

```
[['ዘላለም', 'በባህር', 'ዳር', 'ዩኒቨርሲቲ', 'ተመርቆ', 'መምህር']]
```

```
[34]
w2v = Word2Vec(texts, size=128, min_count=1, workers=12)
print(w2v)
```

```
WARNING:gensim.models.base_any2vec:under 10 jobs per worker: c
Word2Vec(vocab=4325, size=128, alpha=0.025)
```

BiLSTM sequential model multi-label dense 26

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 50, 100)           3000000

 bidirectional (Bidirectiona  (None, 64)               34048
 l)

 dense (Dense)               (None, 26)                1690

=================================================================
Total params: 3,035,738
Trainable params: 3,035,738
Non-trainable params: 0
_________________________________________________________________
```

LSTM sequential model multi-label dense 26

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 128)         553600

 lstm (LSTM)                 (None, 32)                20608

 dense (Dense)               (None, 26)                858

=================================================================
Total params: 575,066
Trainable params: 575,066
Non-trainable params: 0
_________________________________________________________________
```

BilSTM sequential model multi-label dense 26 training the model

```
Epoch 1/5
405/405 [==============================] - 103s 218ms/step - loss: 0.2758 - acc: 0.1883 - val_loss: 0.2399 - val_acc: 0.2808
Epoch 2/5
405/405 [==============================] - 80s 196ms/step - loss: 0.2091 - acc: 0.3907 - val_loss: 0.1830 - val_acc: 0.4064
Epoch 3/5
405/405 [==============================] - 78s 192ms/step - loss: 0.1566 - acc: 0.4938 - val_loss: 0.1420 - val_acc: 0.5025
Epoch 4/5
405/405 [==============================] - 79s 194ms/step - loss: 0.1189 - acc: 0.5488 - val_loss: 0.1128 - val_acc: 0.5123
Epoch 5/5
405/405 [==============================] - 78s 193ms/step - loss: 0.0910 - acc: 0.5710 - val_loss: 0.0915 - val_acc: 0.5345
```