Named Entity Recognition for Wolaytta Language Using Machine Learning Approach

By: Biruk Sidamo Sina

A Thesis Submitted to

Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September 2021

Adama, Ethiopia

Named Entity Recognition for Wolaytta Language Using Machine Learning Approach

By: Biruk Sidamo Sina

Advisor: Tilahun Melak (PhD)

A Thesis Submitted to

Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September 2021

Adama, Ethiopia

Declaration

I hereby declare that this Master Thesis entitled "Named Entity Recognition for Wolaytta Language Using Machine Learning Approach" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

_____	_____	_____
Name of the Student	Signature	Date

Recommendation

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "Named Entity Recognition for Wolaytta Language Using Machine Learning Approach" prepared under my/our guidance by Biruk Sidamo Sina submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering. Therefore, I/we recommend the submission of the revised version of the thesis to the department following the applicable procedures.

_____	_____	_____
Advisor	Signature	Date

Approval Page

I/we, the advisors of the thesis entitled "Named Entity Recognition for Wolaytta Language Using Machine Learning Approach" and developed by Biruk Sidamo Sina hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Biruk Sidamo Sina have read and evaluated the thesis entitled "Named Entity Recognition for Wolaytta Language Using Machine Learning Approach" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master's of Science in Computer Science and Engineering

Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENTS

Foremost, I would like to thank Jesus Christ, our Lord and Savior, for giving the wisdom, strength in exploring things, for the guidance in helping surpass all the trials that I encountered and for giving determination to pursue my study and to make this study possible.

Then, I would like to express my profound gratitude to my advisor Dr. Tilahun Melak, his patience, enthusiasm, co-operations and suggestions made me present this research work to produce in the present form. The door to Dr. Tilahun office was always open at the time I ran into a trouble spot about my research.

Next, I would like to thank staffs of Wolaytta Wogeta Radio Station, Wolaytta Fana Broadcasting Corporation Radio station, Wolaytta Language and Literature Department for their invaluable cooperation in data gathering phase of this study. Especially Mr. Alemayehu Waro from Wolaytta Language and Literature Department in Wolaytta Sodo University, without his passionate participation in data collection and prepared dataset approval, this study could not have been successfully conducted.

A doubt of gratitude is also owed to Wolaita Sodo University for sponsoring me, School of Informatics colleagues specially Dr. Tewodros Abebe, Ms. Birhanesh Fikre for their unsaved cooperation for the completion of this piece of work.

Also, I would like to extend my heartfelt gratitude to Data Science Special Interest Group Members and fellow MSc student you all became my inspirations by keeping me harmoniously. Especially, I am in doubt of gratitude for my friends Mr. Girma Assefa, Mr. Temesgen Tadesse and Mr. Yordanos Athinafe you guys played great role in peer reviewing my thesis work.

My last but not least deepest and sincerest gratitude goes to my loving parents for their moral encouragement as well as their prayers. My father Sidamo Sina (Gashe), my stepmother Terefech Tsona, my elder brother Bereket Sidamo you all are such a great family and my icons. Thank you for making all hardships I had to go through easier.

Dedication

Wholeheartedly I would like to dedicate this thesis to my beloved families, who have been source of my inspirations.

Table of Contents

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

FIGURES **PAGE**

ABBREVIATIONS AND ACRONYMS

CoNLL	Conference on Computational Natural Language Learning
CRF	Conditional Random Field
FBC	Fana Broadcasting Corporation
IE	Information Extraction
ML	Machine Learning
MUC	Message Understanding Conference
MUC-6	Message Understanding Conference-6th round
NER	Named Entity Recognition.
NEs	Named Entities
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Part-of-speech
RF	Random Forest
SVM	Support Vector Machine
WFBCRS	Wolaytta Fana Broadcasting Corporation Radio Station
WLLD	Wolaytta Language and Literature Department
WNER	Wolaytta Named Entity Recognition
WSGI	Web Server Gateway Interface
WWRS	Wolaytta Wogeta Radio Station

ABSTRACT

Currently, the overcrowded nature of digital data has resulted in the challenge of extracting relevant structured information. Information extraction techniques were introduced to solve such a bulky process of searching for a relevant query. One of those Information Extraction techniques known by extracting proper names from unstructured text is Named Entity Recognition. Named Entity Recognition is a significant task to identify named entities from large documents. Significant researches have been conducted on NER for well-studied languages like English. However, it was in infant stage for Ethiopian languages. The objective of current study is to develop Named Entity Recognition for Wolaytta Language. Wolaytta language is morphologically rich, but highly disadvantaged in terms of computational linguistic resources. In this study, we have collected data from three sources Wolaytta Wogeta Radio Station, Wolaytta Fana Broadcasting Corporation Radio Station and Wolaytta Language and Literature Department. Therefore, newly labeled dataset that have 16420 words is used for the study. Current work focused on exploring three main Named Entities Person, Location, and Organization from unstructured Wolaytta text. Word-level feature extraction and DictVectorizer are used for feature engineering task. Then on extracted feature representation, we have employed three machine-learning algorithms, Conditional Random Field, Support Vector Machines and Random Forest to classify Named Entities to their predefined classes. We have undertaken several experiments to determine best performing model. Conditional Random Field model is best performing with 93.8%, 94.9% and 91.4% precision, recall and f1-score respectively over other classifiers. After determining a well-fitted model with our problem, we have examined the combination of feature sets to know which word-level feature combination has more influence in recognizing Named Entities from Wolaytta text. We dropped one feature at a time, the more decrease in model's performance, the more influential that feature is. Therefore, the experiment result shown us that dropping Part-of-Speech tag decreased in 2.2%, 1.2% and 0.8% precision, recall and f1-score and word-shape features resulted in decrease 1.6%, 2.6% and 0.2% respectively from baseline performance. However, applying suffix feature shown less effect in our model. Furthermore, beyond current model building our newly prepared dataset can take vital allotment for future researchers in the area.

Keywords: *Named Entity Recognition, Wolaytta Language, NLP, Machine Learning, Low Resource Language*

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

Nowadays, extracting relevant information from unstructured electronic data have become challenging issue for institutions like news agencies, medical institutions, educational institutions and others too (Singh, 2018). According to Ana et al., data is now digital capital with the explosion of information in the form of news, corporate files, medical records, government documents, court hearing and social media; everyone is flooded with information overload (Ana, 2007). However, the amount of accessible information would not be of much useful if there were no suitable techniques to process it and extract knowledge from it. Overcoming the above problem became easily possible with the help of Information Extraction techniques. Information Extraction is vital pre-processing technique that significantly increases the text mining potential (Ipalakova, 2010). Moreover, one of core components of information extraction is Named Entity Recognition.

At the start, the word Named Entity was coined in Sixth Message Understanding Conference (MUC-6) in United States of America. The aim of conference was to assess research on the automated analysis of military messages that have textual information. Message Understanding Conference was focused on Information Extraction duties at that time, inclusively extracting structured information about company operations and defense-related activities from unstructured text, such as newspaper articles as stated by (Nadeau, 2006).

Hence, Named Entity Recognition became hot Natural Language Processing (NLP) research problem studied for years. Ahmed et al. stated, as NER is sub task of information extraction that locates and categorize named entity mentions in unstructured text into pre-defined classes like person names, locations, organizations, time expressions, quantities, monetary values and percentages (Ahmed, 2010). Moreover, it was considered as one of the most crucial and important preprocessing step in research seeking areas like question answering, entity linking, or machine translation.

The current study attempted to explore the Lexical level of the language by focusing on detecting and categorizing Named Entity mentions in Wolaytta language[1]. Wolaytta language is one of omotic languages. Omotic languages are a group of nearly 30 languages spoken in Ethiopia's southwest near Omo River. Among these 28 omotic languages are divided into two sub-families of northern and southern (Girma Yohannis, 2018). Wolaytta is grouped as northern omotic language spoken in Ethiopia around Wolaytta Zone and other regions of Ethiopia's Southern Nations, Nationalities, and People's Region. The language has more than 3.3 million native and dialective speakers. Currently, the language is serving as medium of instruction throughout education institutions like primary and secondary schools and it is being offered in Wolaita Sodo University as a program. Moreover, the educational institutions, mass medias, health sectors are growing and in need of the language's computational resources.

Even though, the language is morphologically enriched, it is highly disadvantaged language regarding computational linguistic resources. Unavailability of structured corpus is challenging to carry out any NLP research on the language. Since, to come over such problem the current study have inevitable role. We have also passed such resource unavailability cross bar, thanks to the obstacle it have initiated us to come up with our own well-structured dataset. Here, data collected from different sources were prepared and preprocessed to attain our desired NER model-building goal.

Although, NER models might come in three different approach as rule-based, machine learning and hybrid approaches. The rule-based approach is based on handcrafted local language rules. The drawback of rule based approach includes; require the use of linguistic expert to determine all of the rules, tedious task of designing those rules, lack of portability, as well as high maintenance cost when new rules are introduced and old rules are withdrawn. The machine-learning approach extracts some features from prepared corpus with NEs and learns by those feature sets. The last, hybrid is known by incorporating the above two approaches.

[1] Wolaytta language (also known as Wolaytta Doona by native speakers) is one of Omotic languages spoken in southern part of Ethiopia.

The current study was conducted with machine learning approach. As Belay et al., stated machine learning approach is compact, adaptable, and easy to maintain (Belay, 2014). To achieve the objective of the study, data was collected from three different sources and preprocessed in machine understandable form to feed machine-learning algorithms. The machine learning algorithms selected in this study have been identified by their current state-of-art behavior. We have used three algorithms Support Vector Machine, Random Forest and Conditional Random Field. The above algorithms are chosen after comparing their preference for current classification problem. Conditional Random Field is selected as it considers context at the time of prediction. SVM is selected as it linearly classifying behavior for good generalization performance. Random Forest is also chosen by its overfitting limiting behavior. Finally, best-performed model is chosen as Wolaytta NER task.

In this study, we have used Word-level features with DictVectorizer in our feature engineering technique. Then some of our feature sets were combined to check their effectiveness in the language and for determining purpose of best features. Even Named Entity classes are numerous our model's capacity is limited to classifying of three entities namely Person names, Organization mentions and Location mentions were our intended Named Entities. Finally developed NER model have vital role in organizations that work with Wolaytta language in their daily activities, besides it would be applicable in future research in higher NLP studies of the language.

1.2 Motivation

We have motivated to undertake the research to assist the growing news organizations in Wolaytta by developing easy to use Named Entity Query model. Since they play with various Named Entities in their day-to-day activities.

Moreover, resource poorness of the current language also fallen us upon the idea to carry out the study to resolve computational linguistic issue of the Wolaytta language. Obviously, in Ethiopia many languages are disadvantaged i.e., linguistic computationally under-resourced (SIKDAR, 2017). When there is unavailability of annotated corpora, name dictionaries, good morphological analyzers, POS taggers and treebank certain language is said low-resourced (Michael Franklin, 2019).

In addition as research work of (Tewodros A., 2018) stated as the current language is one of Ethiopia's omotic languages with almost no computational resources. Since, the objective of carrying out a research was to solve a particular real world problem; the accomplishment of this research might advance one-step forth the language resource by making current dataset publicly available for future researchers.

1.3 Statement of the Problem

The explosion of electronic data is being produced in organizations that use Wolaytta language as a medium for their daily activities like educational institutions and Social Mass Medias in Wolaytta. There is crowdedness of digital data generated daily in these institutions. Though they are facing challenge of extracting structured and relevant information from those electronic data. Moreover, lack of information extraction techniques like Named Entity Recognition have been bottleneck for their daily activities. Even Named Entities comprised majority of their document's content while holding valuable information, there is no named entity extracting technique developed yet. Therefore, recognizing such Named Entities from unstructured text and classifying them to predefined categories would ease the information extraction need of those institutions

Although it have been tedious task for the employees of the above institutions to extract and answer "who" did "what" and "where" questions. Hence, those institutions are in need of NER platform for answering the above problems. Therefore, the current developed NER model will solve the above stated problems easily and efficiently.

Various Named Entity Recognition researches were carried out on Ethiopian languages such as Afaan Oromo (Sani, 2015) and Amharic (Belay, 2014). However, it is quite difficult to follow the identical NER pattern of those local languages directly for Wolaytta. As, most Ethiopian languages are enriched with different morphological structure and require independent NER model. The current study would develop independent NER for Wolaytta.

Beside the morphological variation, we have gone through various related works. There are some research gaps identified from those related works that need to be handled by the current study. For instance researchers (Ahmed, 2010) and (Alemu, 2013) developed NER model for Amharic. However, usage of limited context size, inconsideration of acronymic NEs, evaluation with small NE instances and poor model performance were identified gaps from these works. On the other hand (Legesse, 2012) and (Sani, 2015) have conducted on NER for Afaan Oromo language. They have developed good model performance, but usage of hand crafted rules that may lead to inadaptability with new NEs and using only one window size were limitations identified from those studies. The current study considers increased feature sets, will handle acronymic NEs (as NEs may come in acronyms) with increased context size to solve limitations from related works.

Though the aim of this study is to recognize Named Entities from newly prepared Wolaytta dataset. The data was gathered from three different sources Wolaytta Language and Literature Department, Wolaytta Wogeta Radio Station, Wolaytta FBC Radio Station and prior researches. Then has to pass some of preprocessing techniques, used for our problem at hand.

1.4 Research Questions

This research tries to answer and address the following research questions:-

RQ1: Which machine-learning algorithm can perform best for Wolaytta NER?

RQ2: What language dependent and language independent features can be used for Wolaytta NER model?

RQ3: Which feature combination can better perform for Wolaytta NER model?

1.5 Objectives of the Study

1.5.1 General Objective

The general objective of the research is to design and implement Named Entity Recognition Model for Wolaytta Language using Supervised Machine Learning Approach.

1.5.2 Specific Objectives

The Specific objectives of the study are:

- ✓ To review existing literatures on NER for various languages.
- ✓ To collect corpus of Wolaytta language.
- ✓ To explore named entities representation in Wolaytta language
- ✓ To select better performing Machine Learning algorithm for Wolaytta language NER.
- ✓ To design optimal NER architecture for the language.
- ✓ To test and evaluate the developed feature model performance.

1.6 Significance of the Study

The developed NER model will serve as an input for higher NLP applications like information retrieval, question answering, machine translation in the language

Furtherly, the current study can be applicable for institutions that work with the language. The thriving Mass Medias in Wolaytta can also use the model since news have various NEs. As, extracting certain relevant information from large content of news can be laborious task for news providers. Therefore, we can utilize current NER model for retrieving people, organizations and places mentions on news.

1.7 Scope and Limitations

1.7.1 Scope of the Study

The current study focuses on introducing NER model for Wolaytta using machine-learning approach. Because of unavailability of Named Entity tagged dataset for the language preparing new dataset is required. To achieve the objectives of the study, our newly built dataset will be preprocessed in machine understandable form by feature extraction technique. The extracted feature representation would be pass to CRF, SVM and RF machine-learning algorithms.

The developed named entity trained models are tested and evaluated with performance evaluation metrics. Finally, better performed model with feature combinations were used for our NER purpose.

1.7.2 Limitations of the Study

The current study has below enlisted limitations due to time and resource constraints:

- ♣ This study is limited to only three named entities namely Location, Person and Organization.
- ♣ Sub categories within each named entity are not considered in the current study.
- ♣ The current study is carried out in sentence level NE recognition, but it does not considered large document level NEs.

1.8 Beneficiaries of the Study

The study's target beneficiaries are:

- ♣ **Wolaytta Mass Medias**: Currently, in Wolaytta there are various Mass Medias including Wolaytta TV, Wolaytta FBC Radio Station, Wolaytta Wogeta Radio Station and Wolaytta Times that are now disseminating public information. They will benefit from current study for their news entity extraction.
- ♣ **Future Academicians:** future academicians can benefit from the study for their referencing purpose. Moreover, other institutions in the domain area that play with NE mentions can benefitted from the study.
- ♣ Institutions that use the language's data in digital form can also benefit from the study.

1.9 Thesis Organization

Introduction (Chapter 1): This chapter elaborates introductory section and deep insight about study. Introduction, problem statement initiated us to carry out the study, research questions, objectives of the study, scope and limitations and final application the study are covered inside it.

Literature Review (Chapter 2): The literature review, principles related to named entity identification, NER methods used so far, NER feature sets, Wolaytta Language structure, and related work are addressed in the section.

Research Methodology (Chapter 3): This chapter deeply elaborates the methodology we have followed to solve Wolaytta NER problem. It overviews research method used, the behavior of the dataset including data collection, corpus annotation and preprocessing procedures we have used. Also software and hardware tools, performance evaluation metrics and features sets are raised throughout this section.

Design of Wolaytta NER (Chapter 4): In this chapter, we have clarified the architecture of WNER. In addition, main processes in the architecture and their sub components were discussed briefly. In addition, deployment prototype architecture is depicted in this section.

Implementation of Wolaytta NER (Chapter 5): This chapter elaborates on the implementation part of the current study. Different libraries and python toolkits used for our implementation were enlisted and described in detail.

Results and Discussion (Chapter 6): In this chapter, we have attempted to describe Results and Discussion of the study. Firstly, Results obtained from our experimentation via specified three algorithms were compared. Then, best performed model via its feature sets are discussed in the next section. After all, we have summarized the whole behavior of the study in conclusion section. Finally, we have recommended good path for future researchers in the domain area.

CHAPTER TWO

2. LITERATURE REVIEW

The general overview of Named Entity Recognition is discussed in depth in this chapter. Different methods for Named Entity Recognition, major subtasks in Named Entity Recognition, evaluation matrices, and morphological structure of Wolaytta language were discussed in the sections below. In order to comprehend the complexity of the work to be performed, the section also covers natural language processing fields that are relevant to NER. Finally, we have described morphological structure and word categories of the language we are exploring.

2.1 Information Extraction

The overwhelming heterogeneity of documents and their various forms (structured, semi-structured, and unstructured) presents new challenges and necessitates the introduction of new technologies.Hence, as the amount of online information increases, the availability of stable, scal able IE systems will become a crucial requirement (Rey, 1999). Information Retrieval and Information Extraction are two NLP tasks that provide a possible means of obtaining access to the information found in large volumes of textual data. A consumer can usually access documents from a wide database using information retrieval systems. Hence, IE is the process of extracting structured knowledge from unstructured or semi-structured documents. Author of (Hirpassa, 2017) stated IE as natural language processing (NLP) program that aims to extract structured factual knowledge from unstructured text.

As author of (Konys, 2018) claimed Information retrieval, especially from unstructured and semi-structured sources, is a difficult task. It is forced to read hundreds of textual documents, web pages, and other data to manually dig out the requisite information without an Information Extraction (IE) tool. For instance, a business domain information extraction system might extract the names of firms, goods, facilities, and financial figures related to business activities. Recent activities in multimedia document processing, such as automated annotation and content retrieval from images, audio, or video can also be called IE. Concisely, IE's objective is to extract relevant types of information from natural language text.

For different languages and domains, different researchers use different measures for developing information extracting systems. The research work in in (Hirpassa, 2017) and (Johannes, 2004) mainly categorizes IE into six separate activities. Those are POS tagging, NER, Syntax Analysis, Co-references and discourse Analysis, Extraction Patterns and Bootstrapping.

Hence, Named entities are important in information extraction because they are used to fill some or even all of the slots in the result templates. The quality of named entity identification would ultimately affect the quality of information extraction (Johannes, 2004). We can conclude that, NER recognition was one of sub-tasks of information extraction system. Below we have elaborated in detail.

2.2 Named Entities

NE is a word or phrase that distinguishes one object from a group of others with similar characte ristics. The majority of NEs are proper nouns, such as a person's name, an organization's name & location's name. For instance, the sentence "*The American president Joe Bayden visited Ethio pia*" has three NEs: America and Ethiopia are countries, and Joe Bayden is an individual.

The sixth Message Understanding Conference introduced the idea of NE, which is still used in IE technology. In fact, the MUC conferences made a major contribution to this field's research. It acted as a benchmark for named entity systems that were used to perform a variety of data extraction tasks as stated in research works (SIKDAR, 2017) (Sani, 2015) (Legesse, 2012) (Chinchor, 1998) (Jing Li, 2018).

As work of (Sani, 2015) (Demissie, 2017) (Legesse, 2012) the sixth message understanding conference divided named entity labels into three classes. The following are the categories:

- ENAMEX (named entities): used for individual, organization, and location entities
- TIMEX (Temporal Expressions): used for date, time entities
- NUMEX (Number Expressions): used for money, percentage, quantity entities.

2.3 Named Entity Recognition

In every natural language, nouns play a very crucial role for information extraction. A subcategory of noun is proper noun. They reflect the names of individuals, places, and organizations, among other items. NER is a task of detecting and then recognizing proper nouns in a text and categorizing them into classes such as person, place, organization, and others (Gitimoni Talukdar, 2014). Tasks like Knowledge Extraction (IE), Query Answering, and Automated Summarization (AS) all involve the identification and labeling of Named Entities (NE) in document.

Identification of proper nouns and further classification of these proper nouns into a collection of classes such as individual names, place names, organization names, and miscellaneous names ar e two stages of NER. In NLP, named entity identification is a vital activity (Jason P.C. Chiu, 2016). The name NER system was first coined at MUC-6, but the system has since been used under various names that are domain specific (extract and recognize company names). After the growing need to extract specific information from a large corpus, the term NER was coined. The most important task in a NER system is defining the right tag for an entity; there are conventions that define tagging techniques for a named entity in a corpus.

Most researchers working on NER are currently attending the Message Understanding Conferences-6 (MUC-6) and MUC-7, as well as the Conference on Computational Natural Language Learning (CoNLL) (Alemu, 2013).

The author of (Shaalan, 2014) clarified NER is a task that intends to detect, extract, and automatically classify named entities in open-domain and unstructured documents, such as newspaper articles, into predefined classes or categories.

2.3.1 Message Understanding Conferences (MUC)

The Message Understanding Conferences (MUC) is a series of conferences coordinated by the N ational Institute of Standards and Technology and the Advanced Research Projects Organization of the United States Department of Defense. MUC's seven IE conferences are thought to have ha d a major impact on the development of NLP science, especially in IE. The conferences concentr ated on the languages of English, Chines, Japanese, and Spanish. ENMAX, TIMEX, and NUME X are the three conference entities that have been identified.

MUC's were held at various times to enhance automated text extraction, which includes taking measurements to assess software systems' ability to correctly recognize and extract information f rom text documents. For example, in MUC6, there has been a growing interest in NER, and a sig nificant amount of effort has gone into research (Sani, 2015) (Shaalan, 2014).

Furthermore, during the MUC7 conference in 1998, date and time entities, as well as multilingua l named entity recognition, were implemented, and an assessment of the Named Entity Recogniti on task, a coreference task, and three tasks involving extracting information into templates was c arried out (Chinchor, 1998).

2.3.2 Applications of Named Entity Recognition

The pervasiveness of named entities, as shown by the high level of occurrence and co-occurrence of named entities in corpora, is one clear explanation for their significance. Since, NER makes it easy to recognize key elements in a document, such as people's names, locations, labels, and monetary values. At the time of dealing with large datasets, extracting the key entities in a text can help sort unstructured data and detect essential information. The authors of (Demissie, 2017) and (Shaalan, 2014) stated some of the applications for which NER is useful as:

Information Retrieval: The process of defining and retrieving relevant documents from a collection of data based on an input query is referred to as IR. According to the author, NEs appear in approximately 71% of all search engine queries.

Question Answering: This is similar to information retrieval, but the findings are more sophisticated. A Question Answering System accepts questions as feedback and responds with thoughtful and insightful responses. The named entity recognition is also used during the query analysis to recognize NEs in the question, which will assist in later identifying relevant documents and constructing the answer from relevant passages. Furthermore, since the answer to several factoid questions requires NEs, Question Answering Systems may benefit greatly from NER.

Machine Translation: The process of automatically translating a text from one natural language into another is known as machine translation. When determining which parts of a NE should be meaning-translated and which parts should be phoneme-transliterated, NEs need special consideration.

Text Clustering: Clustering search results will take advantage of NER by rating the clusters based on the number of entities in each (Demissie, 2017).

Navigation Systems: These systems, which enable us to navigate using digital maps, have become increasingly important in our lives. Directions, information about nearby locations that may be connected to other online services, and traffic conditions are all offered. Points of interest are NEs that are stored in a database along with their spatial coordinates in these systems.

2.3.3 NER Architecture

According to (Belay, 2014) (Legesse, 2012) (Martin, 2009) regardless of whether it is constructed using a rule-based approach or an automated machine learning approach, a standard named entity recognizer has four core elements. Figure 2.1 portrays the architecture of a typical NER model. Tokenization, morphological and lexical processing, identification and classification are the key components.

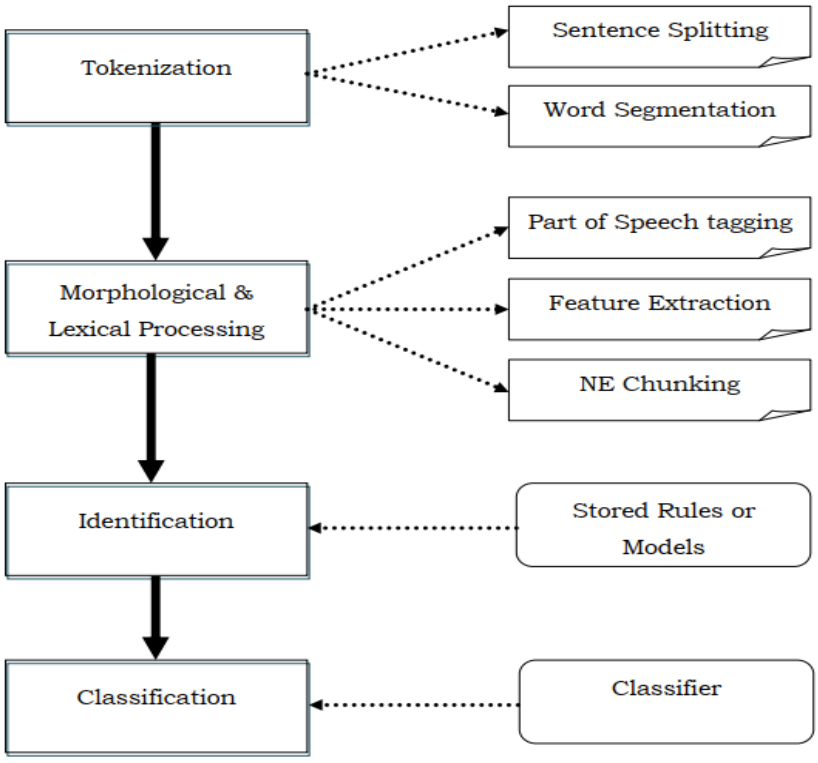


Figure 2. 1 **NER architecture**

From the above depicted architecture we can define subtasks as:

Tokenization: As method of separating a text, paragraph, or sentence into meaningful units from a string of words or characters.

Morphological and Lexical Processing: In this stage, the token that was previously defined is f urther processed, for example, to associate features such as the POS tag or the token's boundary, whether it is within or outside of the Named entity.

Identification: Using either learning models or predefined laws, this step involves detecting nam ed entities.

Classification: After the named entities have been defined, this step will classify them into their acceptable forms.

2.3.4 Approaches of Named Entity Recognition

The automated extraction and classification of named entity systems are categorized according to the approach they have used, as is the case for most Natural Language Processing systems.

Many research works including (Ahmed, 2010) (Belay, 2014) (Demissie, 2017) (Legesse, 2012) (Padmaja Sharma, 2014) (Gitimoni Talukdar, 2014) stated those approaches as Rule-based, Machine-Learning, and Hybrid are the three basic forms. Below we elaborated one by one.

2.3.4.1 Rule-based Approach

Linguists manually write and describe a set of rules in this approach. Handcrafted rule-based NER systems derive names and entities from a vast number of human made rule sets. These models work better in limited domains and are capable of detecting complex entities that l earning models struggle with (Legesse, 2012). Rule-based approaches fails to satisfy the demands of portability and robustness, and finding the rules based on which the method is supposed to provide the best results necessitates a great deal of linguistic expertise. In the MUC-6 and MUC-7 competitions, rule-based structures performed admirably (SOLOMON T., 2015).

Also, the issue with this approach is that manually making rules is time consuming, costly and extremely difficult to involve all of the rules required for recognition and classification. In addition to that costs of maintaining rule-based structures are extremely high. However, once all of the necessary rules have been identified, the system can detect and identify name entities with ease (Ahmed, 2010) (Gitimoni Talukdar, 2014) (Padmaja Sharma, 2014).

2.3.4.2 Machine Learning Approach

This approach of NER allow us the use of learning algorithms that involve large tagged data sets for training and testing. In order to create statistical models for NE prediction, ML algorithms use a selected set of features derived from data sets annotated with NEs.

The author of (Shaalan, 2014) stated as machine learning approached NER systems have the advantage of being easily adaptable, portable and updatable with minimum time and effort. As a result, these methods are very common and advantageous compared to Rule based NER because they are less costly to maintain and can easily be extended to new languages and domains.

Also authors of (Padmaja Sharma, 2014) clarified as major benefit of machine learning (ML) is that it is trainable and adaptable to a range of languages and domains. Furthermore, the cost of maintenance is lower than with the rule-based approach. The main aim of the ML approach is to use statistical models to classify proper names in order to recognize them. Hidden Markov Models (HMM), Conditional Random Fields (CRF), Support Vector Machines (SVM), and Maximum Entropy are instances of classification algorithms used in NER.

This approach can further divided into three categories namely supervised learning, unsupervised learning, and semi-supervised learning as work on (Belay, 2014).

1) Supervised Learning (SL): The supervised machine learning algorithms are currently the most common tool for solving the NER issue. They are referred to as supervised because they learn from well-labeled data that shows them what to predict. Hidden Markov Models, CRF and SVM are instances of this type.

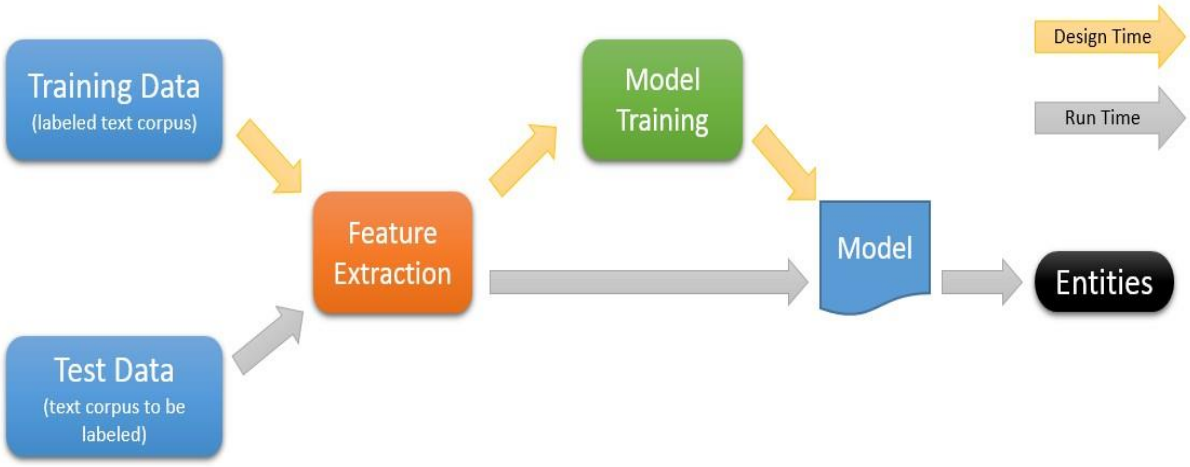


Figure 2. 2 NER workflows[2]

2) Un Supervised Learning: The system is given unlabeled data in this method, and the algorithms' aim is to find a representation for the data. These representations can then be used for data compression, classification, decision-making, and a number of other applications. For NER, unsupervised learning is not a common process. Expected Maximization and K-means are two examples.

3) Semi-supervised Learning (SSL): Now a days Semi-supervised learning has gotten a lot of attention in the pattern recognition and machine learning fields (P. K. Mallapragada, Nov. 2009).

It was believed that as modern method for machine learning that has come close to approach in terms of success in the domain of NER. As (B. Aryoyudanta, 2016) clarified the small amount of labeled data for the training phase is a concern when using machine learning in a Named Entity Recognition (NER) system. Unlabeled data, on the other hand, is widely accessible from a number of sources, unlike labelled data, which is limited. This makes it possible to solve the NER system problem using a semi-supervised learning method. The most popular SSL technique is called "bootstrapping," and it requires a small amount of supervision, such as a collection of seeds, to get the learning process underway.

[2] https://blog.vsoftconsulting.com/blog/understanding-named-entity-recognition-pre-trained-models

2.3.4.3 Hybrid Approach

This approach is the result of integrating rule-based and machine-learning techniques, as the name suggests. It is generated by incorporating the best features of both rule-based and machine learning approaches. Research work of (Padmaja Sharma, 2014) clued as this method preferable when there is a shortage of data to construct a NER model.

In at least two instances, a hybrid approach to NER can be used. To begin, the NER's rule-based component can be included as part of the feature space, providing more contextual knowledge about a given word to the ML component during the learning process. Second, a rule-based component may be used to post-process and disambiguate data that has already been categorized (Belay, 2014).

2.3.5 Feature Space for Named Entity Recognition

The properties or characteristic attributes of words built for consumption by a computational framework are features in NER (Gamback, 2018). This procedure starts by translating the set of words (tokens) classified into a set of feature vectors belonging to a feature space that are fed as input to the text classifier. The feature vector representation is a text abstraction that typically assigns one or more Boolean or binary values (such as whether a word is capitalized), numerical values (word length), and nominal values to each word. The choice of features considered by a classifier is a critical issue in the ML approach, and it can have a major effect on the model performance.

As work on (Shaalan, 2014) attempted to state the main aim of selecting features is to find a clear connection between a NE and one or more combined features such that generalizations can be explored throughout the collection of features. Iterative tests are carried out to better understand the effect of various combinations of the selected features on the NER role. The feature space becomes high dimensional when all feature values and their combinations are chosen. For the recognition mission, not all features are equally relevant. In order to find the best feature set for a NER model, even the set of selected features must be evaluated.

Many researches including (Belay, 2014) (Alemu, 2013) (Shaalan, 2014) (DAWED, 2020)exemplified a hypothetical NER system might assign three attributes to each word in a document.

1. Boolean attribute with the value true if the word is capitalized and false otherwise;
2. Numeric attribute corresponding to the length in characters of the word;
3. Nominal attribute corresponding to the lowercased version of the word.

For instance, in our case of above scheme the sentence “Amerikka biittan korona vayressiyaa oyqettida assatu qooday daridooga Presidante Donaldi Trampi qoncisiis” by omitting the punctuation, the above sentence can be represented by the following feature vectors:

<true, 8, “amerikka”>, <false, 7, “biittan”>, <false, 6, “korona”>, < false, 11, “vayressiyaa”>, <false, 9,“ oyqettida”>, <false, 6, “assatu”>, <false, 6, “qooday”>, <false, 9, “daridooga”>, < true, 10, “presidante”>, < true, 7, “donaldi”> < true, 6, “trampi”>, , < false, 9, “qoncisiis”>

This feature engineering technique can be operated only with languages that use capitalization for named entities, Hence Wolaytta language can also benefit from definition as it follows capitalization for NEs.

The research work on Amharic (Belay, 2014) (DAWED, 2020) took the same approach as the previous example, but since the language lacks capitalization, they added additional features such as whether the word is at the beginning or not, and whether the word is a noun or a proper noun.

They used the following three attributes:

1. A Boolean attribute that determines whether a given word is the start of a sentence or not.

2. A numeric attribute that keeps track of how many characters are in a given word.

3. A Boolean attribute that determines whether a word is a noun, noun phrase, or not.

Many research works like (Ahmed, 2010) (Belay, 2014) (Sani, 2015) (Demissie, 2017) (Legesse, 2012) argued as NER features are most widely classified into three function groups, according to their work: word-level features, list look-up features, and document and corpus features.

2.3.5.1 Word-Level Features

These features pertain to the characters that make up a word as well as its grammatical sense. First letter capitalized, numeric, punctuation, part of speech, morphology, and individual characters in a given word are only a few examples.

Table 2. 1 Word-level features

Features	**Description with examples**
Case	▪ Whether the first letter of a word is capitalized, ▪ All letters are uppercased , or ▪ The word consists of mixed cases
Punctuation	▪ Ends with period, has internal period (e.g., St., I.B.M.) ▪ Internal apostrophe, hyphen or ampersand (e.g., O'Connor)
Digit	▪ Digit pattern (e.g., date, percentage, interval) ▪ Cardinal and Ordinal ▪ Roman number ▪ Word with digits (e.g., W3C, 3M)
Character	▪ Whether special character are used like Greek letters or mathematical symbol ▪ Possessive mark, first person pronoun
Morphology	▪ prefix and suffix morphemes, stem, and ▪ Common ending word.
Part of speech	▪ Whether the word is a noun, verb, adjective, and etc.
Character length	▪ Number of characters in a word ▪ Number of characters of a prefix and suffix.

2.3.5.2 List look-up features

In NER, lists are the most powerful functions. The words "gazetteer," "lexicon," and "dictionary" are often interchanged with "list." List inclusion is a way to express the relation “is a” between words (DAWED, 2020).

Table 2. 2 List lookup features

Features	**Description with examples**
General List	This list includes list of :- ▪ stop words (function words) ▪ common abbreviations and ▪ Capitalized nouns (e.g. Sunday, September and the like).
List of entities	These are lists combined of the entities that the NER is recognizing. For example ▪ Person name lists, ▪ organization name lists ▪ Location name lists etc.
List of entity clues	These lists can include trigger words found in most entities. ▪ Particular clue words in organization. ▪ Person name title, prefix of person name ▪ Location particular words

2.3.5.3 Document and Corpus Features

Document features are described in terms of both the content and the structure of the document. Properties can also be found in large collections of documents (corpora). These features cover the function of a given word through a text or corpus. Many occurrences and events, entity co-reference and alias, and statistic for multiword limits are just a few of the functions as we below depicted in table (Sani, 2015).

Table 2. 3 Document and corpus features

Features	**Description with examples**
Multiple occurrences	▪ Other entities in the context ▪ Uppercased and lowercased occurrences ▪ Anaphora, co-reference
Local syntax	▪ Enumeration, apposition ▪ Position in sentence, in paragraph, and in document
Meta information	▪ Uri, Email header, XML section ▪ Bulleted/numbered lists, tables, figures
Corpus frequency	▪ Word and phrase frequency ▪ Co-occurrences ▪ Multiword unit permanency

In the context of any machine learning technique, feature estimation plays a critical role in the Named Entity Recognition task. Many features can be found in Ethiopian languages by considering the various combinations of available context words.

In addition the research work on (Gitimoni Talukdar, Supervised Named Entity Recognition in Assamese language, 2014) stated that various features used in Named Entity Recognition. Those features used in NER task are given as follows:

a) First word: This feature checks to see whether the word appears as the first word of the sentence or not, since the first word in most cases appears to be a named entity that appears often in the subject position.

b) Context word feature: The next and previous words of a word are used in the context word feature. This feature provides useful information for identifying named entities.

c) Part of speech information: POS of the surrounding words and the current word will also aid in the recognition of named entities.

d) Word suffix: The current word's fixed or variable length word suffix, as well as the words around it, may be treated as a feature. In the sense that named entities sometimes combine with common suffixes, suffix information is useful.

e) **Word prefix:** Prefixes may be either fixed or variable in length. Prefix information is often useful since named entities frequently combine with common prefixes.

f) **Infrequent word:** This feature is based on the fact that terms that appear infrequently are called entities. By measuring the frequency of each and every word in the training corpus, infrequent words can be identified. Infrequent words are those that appear less frequently than a predetermined cut-off frequency.

g) **Digit features**: These are used to categorize a variety of named entities such as numerical expressions, percentages, and time expressions.

h) **Length of the word:** This is a binary feature based on the fact that named individuals are seldom short terms. It determines whether a word's length is less than or equal to a specific value.

2.3.6 NER Evaluation metrics

Comparing the system's performance to the human-annotated test data is how NER systems are evaluated. The same measures imported from Information Retrieval are used in NER's evaluation schemes. They do, however, calculate these metrics in various ways.

In our case Precision, recall, and a combined metric called F1-Score have all been adopted by information retrieval as standard evaluation metrics. The precision score (equation (2.1)) is a measure of how much of the information returned by the machine is actually accurate, i.e, ratio between the number of NEs correctly identified by the system True Positives (TP) divided by the total number of NEs returned by the system.

The recall score indicates ratio between number of NEs correctly identified by the system (TP) divided by the number of NEs that the system was expected to recognize. (equation (2.2)). The precision and recall scores are diametrically opposed. As a result of this situation, F-measure a combined measure, has been created. It strikes a balance between recall and precision, and this combined metric measure became the NER system's output performance index (equation) (Bayu Aryoyudanta, 2016).

$$Precission = \frac{TP}{TP + FP} \qquad \textbf{Equation 2. 1}$$

$$= \frac{Number\ of\ exact\ named\ entities\ found\ by\ the\ system}{Number\ of\ Named\ Entities\ found\ by\ the\ system}$$

$$Recall = \frac{TP}{TP + FN} \qquad \textbf{Equation 2. 2}$$

$$= \frac{Number\ of\ exact\ named\ entities\ found\ by\ the\ system}{Total\ number\ of\ Named\ Entities}$$

F-measure is the harmonic mean of the above Recall and Precision values.

$$F - measure = \frac{2PR}{P + R} \qquad \textbf{Equation 2. 3}$$

2.4 Related Works

Here we attempted to present some of the NER researches that has been done so far in this section. We have chosen the most relevant and recent ones among them, which are related to our work from local and foreign languages by incorporating approach they followed, Dataset they used, Algorithm they consumed, feature set used and finally model performance found for the respective research work.

2.4.1 Named Entity Recognition for Local Languages

Below we have gone through some of NER works undertaken in Ethiopian languages.

2.4.1.1 Named Entity Recognition for Amharic language

A lot of research work has been conducted and their results established in the area of NER throughout the world and some countable in fingers conducted in Ethiopia as some are below depicted.

Author of (SIKDAR, 2017) conducted NER for Amharic using deep learning using RNN, a bi-directional LSTM model. The developed NER model achieved precision, recall and f1-score 77.2%, 63.4%, and 69.7% respectively with 10-fold cross validation.

Authors (Ahmed, 2010) and (Alemu, 2013) used Machine Learning approached NER task of Amharic using Conditional Random Fields. Their classifiers have been trained on different word and context features by using about 90% of their data for training and 10% for testing.

Ahmed achieved recall, precision, and F1-measure value of 75.0%, 74.2%, and 74.6%, respectively on a 10,405 word subset of the WALTA corpus, of which 961 words including only 96 NEs were used for testing in which he claimed as part-of-speech tagging improved the results, while using word prefixes did not performed well.

Work of (Alemu, 2013), conducted in Amharic NER but used different approach and feature set than the former one, i.e. he proposed supervised machine learning with conditional random field algorithm with corpus size of 13,538 words of WALTA corpus subset from those 1,242 words were used for testing. Features he used were Previous and next word, named entity tag of a word, word pairs, word shape, prefix and suffix features, and stated that context windows of up to two words before and after the current token improved his work as recall to 84.9% % and precision to 76.8%, for an F-score of 80.7%.

Also (Demissie, 2017) attempted to work Amharic Named Entity Recognition Using Neural Word Embedding as a Feature, in his work SVM, J48, random tree, IBk (Instance based learning with parameter k) attribute selected and OneR classifiers algorithms had been tested with word vector features. He also tested with MLP (multi-layer perceptron) deep neural networks, BLSTM and LSTM then F-score achieved was 95.5% using the SVM classifier.

Author (Belay, 2014) used hybrid-learning approach for his Amharic NER work. On a portion of the dataset annotated by (Alemu, 2013). Belay et al., used a combination of decision trees, support vector machines, and hand-crafted rules, but he arbitrarily extended it to get a better balance between the classes: since 9,899 of the 12,196 tokens he used were tagged as non-NEs, he inflated the NE classes so that the training dataset comprised 31,347 instances.

Despite using 40% of the training data for research, he found that a simple baseline method using only part-of-speech information and a binary flag for nominal outperformed his hybrid learning approach.

Authors of (Gamback, 2018) carried out NER for Amharic Using Stack-Based Deep Learning. Here recurrent neural network, a bi-directional long short term memory model, was used to implement a named entity recognition system for Amharic. To predict each token's named entity type, a Conditional Random Fields (CRF) classifier was trained on language independent features and word vectors based on semantic information were created using an unsupervised learning algorithm called word2vec. To assign labels to the words, the deep neural network was fed the predictions, features, and word vectors. This stack-based solution outperformed numerous other deep-learning setups, as well as a baseline CRF classifier, with an F-score of 74.26 %.

Also (DAWED, 2020) used ML approach to study the Amharic NER structure on his research work. The experiments were carried out on a total of 18,191 tokens for training, development, and testing, using the CoNLL2002 format, BIO tagging scheme, and the Ling Pipe library as a tool for improving the NER for Amharic.

He evaluated the output of the proposed Conditional Random Field (CRF) based Amharic Named Entity Recognition (ANER) method in five different types of experiments with different feature sets.

Author (SOLOMON T., 2015) also used the same ML approach. He specified and used a new collection of features in addition to the POS, gazetteer, and unique features to construct a supervised NER classifier from IOB format marked data, and got 94.2 % F-Measure.

2.4.1.2 Named Entity Recognition for Afaan Oromo language

Different researchers carried out NER for Afaan Oromo in different approaches. Work of (Legesse, 2012) proposed Afaan Oromo NER based on a hybrid approach that includes machine learning and rule-based components. Based on CoNLL's 2002 BIO tagging scheme, Afaan Oromo NE corpus of over 23,000 words has been created. Person, location, organization, and miscellaneous are the four NE categories defined and used in the analysis. NE Chunker, Feature Extractor, and Model Builder were three of the system's components.

Position, word-shape, POS, normalization, prefix, and suffix of a token were among the features he used. The parameter of the model was estimated using stochastic gradient descent. He claimed as he obtained an average performance of Recall 77.41%, Precision 75.80% and F1-measure 76.60% in two major experimentations i.e., increasing the amount of the training data and examining the influence of features on that.

Also (Sani, 2015) studied Afaan Oromo NER System using a hybrid approach that includes machine learning (ML) and rule-based components. Parsing, filtering, grammar rules, whitelist gazetteers, blacklist gazetteers, and exact matching are all included in the rule-based section. The ML component consists of two parts: an ML model and a classifier. For the rule-based component, he used the GATE developer platform, and for the machine-learning component, he used Weka. He used a corpus with a size of 27,588 that was generated by (Legesse, 2012). He used a corpus of size 23,000 for training and 4,588 for evaluating the model. He attempted to increase the amount of data consumed by the algorithm and discovered that the average result was 84.12% Precision, 81.21 % Recall, and 82.52 % F-Score. Hence increased the model performance to some extent. The research work on (N.Kannaiya Raja, 2019) also carried out creation of a Named Entity Recognition (NER) method for the Afaan Oromo language presented using a rule-based approach.

2.4.2 Named Entity Recognition for Foreign Languages

Here we attempted to review some of most related NER of foreign languages, but not all.

2.4.2 .1 Named Entity Recognition for Chinese

Different researches have been carried out in the domain of NER for Chinese Language using different approaches below we attempted to summarize some of them.

As work on (Wenliang Chen, 2006) Named Entity Recognition method for Chinese using Machine Learning approach was introduced. The Conditional Random Fields model was used in their research work. They used a training data format that was adapted for Chinese from the CoNLL NER task 2002. The data was displayed in a two-column format, with the character in the first column and the tag in the second. For their NER system, they used basic features, word boundary features, and char features. They conducted the post-processing technique according to the CRFs model's n-best results in order to correct inconsistent results. As they claimed, their final system achieved an F-score of 85.14 %.

Also the research in (Sun, 2017) attempted to solve a Chinese NER task on a postal address corpus, the authors used a modified Conditional Random Field (CRF) model. They use the documented, useful, simpler semantics words and sentences to their actual model as the additional features, as there has been little data about Chinese postal addresses and sections of which are incomplete sentences. They conducted three tests to assess the system's accuracy, and the results showed that their improved algorithm outperforms other traditional algorithms when processing postal addresses.

To deal with Chinese NER, researchers first used a variety of algorithms or methods to extract different and representative features. They barely defined the relationship between words and contexts using POS. Then they created a CRF feature template with all of the features mentioned above. To count the number of different tokens used four arguments: tr, wr, tu, wu, where tr denotes the number of correct recognized tokens, wr denotes the number of wrong recognized tokens, tu denotes the number of correct unrecognized tokens, and wu denotes the number of wrong unrecognized tokens. To test the output of their model they have counted the values of Accuracy, Precision, Recall, and F-Measure during both two listed below experiments.

They used two experiments in their research. There are two types of weight classes: lightweight and heavyweight. The first experiment compares various types of features and feature models used for CNER, with 21 sentences and 306 tokens in our training data, 216 of which are tokens from 62 entities, and 71 sentences and 1201 tokens in our testing data, 835 of which are tokens from 228 entities. The features that worked well in the first experiment using the heavyweight data set are the subject of the second experiment. In 395 sentences, there are 9145 tokens, 6777 of which are entity tokens. To calculate the mean value and standard deviation, they used 4-fold cross validation. Finally, their model performed admirably 83.25% F1-Score.

2.4.2.2 Named Entity Recognition for Indian Languages

Authors of (Nita Patil, 2020) have proposed a simple NER system for the Marathi language one of morphologically rich Indian languages, which is based on a manually annotated NE labelled corpus of 27,177 sentences. The framework implemented a feature function that takes a sentence, the current word's position, the current word's label, and the previous word's label from the training dataset as input parameters and predicts the most suitable NE tag for a word in the sentence based on the learning. The data containing 27,177 Marathi sentences containing 63,236 unique words is loaded to apply their model.

In the study, they begin by constructing a CRF based simple NE Tagger that was trained on a manually tagged data set of 26,462 sentences and tested on a test data set of 715 sentences. Overall named entity identification accuracy was 75.7% (F1-Measure) and average named entity classification accuracy was 75.51 % (F1-Measure) for the proposed NER framework. They reported as system output can have an impact on whether or not data is useful for training, but this is part of the testing process. As a result, a 10-fold cross-validation test is used to determine the Marathi named entity recognizer's accuracy. The annotated data is divided into ten equal parts.

Also author (Gowri Prasad, 2015) proposed a method for Named Entity Recognition in Malayalam using Conditional Random Field Approach, a supervised machine learning approach. They used a corpus of nearly 50000 Malayalam words from the tourism domain. The first training was conducted with a 25000-word training file and then evaluated on a 2000-word test corpus. Part of Speech tagging was also applied to tokenized text in order to assign a category to each word. For each word, chunk tags were also added. The tool they used for this work was the CRF++ 0.58 kit, which is open source. BIO notation was used for NE classification.

Here Parts of speech of the previous two words, current word, and succeeding two words, chunk tags of the previous two words, current word, and succeeding two words, previous two words and succeeding two words as context words, current word, and combinations of each of the features listed are among the features used in the proposed method. Precision, Recall, and F1-score were the assessment parameters used, and the model scored 90.9%, 62.9% 74.35% respectively.

Research work on (Thoudam D, 2009) discussed also the development of a NER framework for Manipuri, a less computerized Indian language using SVM. Here two different models have been developed, one based on the well-known Support Vector Machine and the other on an active learning technique based on context patterns created from an unlabeled news corpus. They used active learning methods as a baseline. To add SVM to the Manipuri news corpus, the major NE tags, namely Person name, Place name, Organization name, and Miscellaneous name, were manually annotated.

The NE tagset has been manually annotated on approximately 28,629 word forms in the news corpus. The SVM-based system is evaluated on a test set of 4,763 word forms after the best feature set is chosen. The features that gave them the best result for the development set were Prefixes and suffixes of the current word with a length of up to three characters, dynamic NE tags of the previous two words, POS tags of the previous two and next two words, Digit information, Word length, Infrequent word. Baseline performed Recall (60.21%), Precision (54.12%), and F-Measure (57.01%), and the final method demonstrated Recall, Precision, and F-Score values 92.32%, 94.03% and 93.17% respectively by utilizing above baseline experiment.

Table 2. 4 Summary of Review of related Works for Local Languages

No	Authors & Year	Title and Corpus size	Algorithms and Features used		Main Findings	Gap
			Algorithm	Features		
1.	(Alemu, 2013)	Named Entity Recognition for Amharic(corpus sized 13,538 words, 1242 for testing)	CRF	▪ Previous and next word, ▪ NE tag of a word, ▪ word pairs, ▪ word shape ▪ prefix and suffix	Achieved F-measure precision and recall with 80.7%, 76.8% and 84.9% respectively	✓ The model works only with context size of two ✓ Acronymic entities haven't handled
2.	(Ahmed, 2010)	Named Entity Recognition for Amharic Language (corpus sized 10,405 words, 961 for testing)	CRF	▪ Word and tag context features, ▪ Part of speech ▪ tags of tokens ▪ Prefix and suffix	Using four scenarios achieved Precision, Recall and F-measure of 72%, 75% and 73.47% respectively	✓ Used small amount NEs for evaluation ✓ Acronymic entities haven't handled ✓ Poor model performance

3.	(Legesse, 2012)	Afaan Oromo NER based on a hybrid approach (corpus sized 23,000 words)	CRF	▪ Word Position ▪ word-shape, suffix of a token ▪ POS, Prefix,	Obtained performance of Recall, Precision F1-measure of 77.41%, 75.80% and 76.60% respectively	✓ Model only works with one context size ✓ Developed model was poorly performing even though the corpus size was high ✓ Acronymic Entities didn't included
4.	(Sani, 2015)	Afaan Oromo Named entity recognition using Hybrid Approach (corpus sized 27,588 words, 4,588 for model evaluation)	J48 as an implementation of C45 for decision trees	▪ POS tag ▪ Word length flag ▪ Capitalization flag ▪ Actual NE tag of the word	Achieved average result of Precision, Recall and F1-score of 84.12%, 81.21% and 82.52% respectively	✓ Used hand-crafted rules that were unadaptable with new instances ✓ Used context size of one

2.5 Wolaytta Language

Wolaytta language structure and background will be discussed in brief through this section.

2.5.1 Background

Wolaytta district is located in Ethiopia's western part. It's 400 kilometers southwest of Addis Ababa. The Wolaytta people refer to their own language as "Wolaytta" (wolaittattuwa in their language) according to WAKASA (WAKASA, 2008). As the authors of (Sottile, 2000) (Alambo, 2010) clarified Wolaytta doonaa (literally "mouth of Wolaytta") and Wolaytta Kaalaa (literally word of Wolaytta) are other names used interchangeably for the language.

Wolaytta is a member of the Omotic language family, which is a subgroup of the Afro-asiatic language phylum (WAKASA, 2008) (Tewodros A. J. N., 2018). It is spoken in the Wolaytta Zone as well as other areas with similar linguistic diversity of Ethiopia's Southern Nations, Nationalities, and People's Region like Gamo[3], Gofa[4] and Dawuro[5]. Wolaytta has been written in the Latin alphabet since the 1940s and has a standardized orthography (Alambo, 2010).

In 1981, a Bible was written in Wolaytta. Wolaytta is an agglutinative language, which means that suffixes can be added to root words to construct new word forms. Using derivational and inflectional morphemes, several word forms can be developed from a single root word as in Table 2.5 below. The morphotactic rules of the language determine the order in which morphemes are added (Tewodros A. J. N., 2018).

Below are instances of morphology (word formations) in Wolaytta language of verb “essa”.

[3] Gamo people located in SNNPR of Gamo zone who were known by their language Gamotso share similar dialect with Wolaytta.

[4] Gofa people similarly located in SNNPR of Gofa zone known by their language Goofatso share common dialect with Wolaytta.

[5] Dawuro people located in SNNPR Dawro zone share common dialect with Wolaytta language.

Table 2. 5 Sample morphology of Wolaytta language

Wolaytta sentence	Root Word	Suffix added	Respective meaning in English
Ta kaamiyaa essaas	**ess-**	-aas	I parked the car
Nu kaamiyaa essida		-ida	We parked the car
Ta kaamiyaa essays		-ays	I park the car
I kaamiyaa essiis		-iis	He parked the car
A kaamiyaa essaasu		-aasu	She parked the car
A kaamiyaa essawusu		-awusu	She parks the car
Nu kaamiyaa essoos		-oos	We park the car
I kaamiyaa essees		-ees	He parks the car

As the agglutinative nature of Wolaytta language many words forms can be derived from specific root word. For example in Table 2.5, the root word is "ess-" that was verb class. Since, word forms "ess-aas", "ess-ida", "ess-ays", "ess-iis", "ess-aasu", "ess-awusu", "ess-oos", "ess-ees" were derived words with respective suffixes. Words like Ta(I), Nu(we), I(He), A(she) in the language denote pronouns they can be substituted with Person names such as:

Tamasgaani kaamiyaa essiis. (Temesgen parked the car i.e, Tamasgaani is person)

Baallota kaamiyaa essaasu. (Ballote parked the car i.e, Ballote is person)

In Wolaytta language suffix endings "**-iis**" and "**aasu**" are verbal endings used to denote male and female actions repectively.

Corppoy ba mayuwaa ***meeciis***. Corppo **washed** his cloth.

Almaaza ba mayuwaa ***meecaasu***. Almaz **washed** her cloth.

2.5.2 Writing system of Wolaytta Language

Wolaytta language uses Latin alphabet and it have its own consonants and vowels sounds. It have generally twenty nine consonant phonemes, of those five of them are combined consonant letters like ch, dh, ph, sh, and zh. There is extremely rare sound written ⟨nh⟩ as described in (WAKASA, 2008) stated as a nasalized glottal fricative as it occurs in only one common noun, an interjection, and two proper names. Additionally Wolaytta has five short and five long vowels. The languages alphabet is characterized by capital and small letters like English alphabet does.

2.5.3 Wolaytta Language Word Categories

Words are comprise fundamental part of a given language. The arrangement of word or their combination depends on the rule or grammar of that language. The combination of those words on the bases of the language gives us sentences. The meanings of those sentences depend upon each word of the sentence and the way they're arranged.

While we came to grammatical categories of Wolaytta language, as work of (Herano, 2015) and (Fikre, 2020) stated words were initially categorized into eleven grammatical categories (Noun, Pronoun, Verb, Adverb, Adjective, Numerals, Preposition, Conjunction, Interjection, Determinants and Punctuation). In addition to above stated grammatical categories the language also uses the combined form of them. Hence, research work of (Fikre, 2020) carried out POS tag for Wolaytta by combining above word classes and identified 26 part of speech tag sets by contributing some features on work of (Herano, 2015) and (WAKASA, 2008) with the help of language experts.

Since our work involves the identification and classification of proper nouns from a given sentence we will see word categories that have contributions to our study.

Nouns

As English and Afaan Oromo, Wolaytta nouns are used to denote name or identify things, people, animals, places or abstract ideas. For instance, Toomasa, Boora, Tophphee, qaala are nouns. In Wolaytta most of the time a sentence begins with a noun, which starts with capital letter as in most Latin languages, does. Most of the time nouns in Wolaytta language can be followed with conjunction endings (-n, -ara, -ppe, -nne, -yo) see below.

Examples of nouns with their POS tag:

Sentence 1: Tophphe bayra dere. Ethiopia is great country.

With POS: Tophphe/NN bayra dere/NN

Sentence 2: Ayalaynne Almaza giya biida Ayele and Almaz went to market.

With POS: Ayalay-nne /NC Almaza/NN giya biida.

Wolaytta nouns can be classified into different classes as English and Afaan Oromo does. These are countable nouns, in countable nouns, collective nouns, common nouns and proper nouns. Since, most of our named entities are proper nouns we focused on them.

Proper nouns: these types of nouns, which are important to our study, are nouns that particularly identify a specific person, organization, location names and others.

Examples of: Person names: Tamasgana, Toomasa, Cuuruqo, Ababa, Ayaala etc. Location names: Amerkkaa , Tophphee, Appirkaa, Wolaytta, Bishooftuu etc. Organization Names: Sodo Pirdee Ketta, Humbo 1ro Xekka Luxxettaa Ketta, Bodittee Aysuwa Keeta, Adaama Siencenne Teknolojje Univurshaa etc.

Above proper nouns can come along with different suffixes and conjunction bound morpheme In the implementation part of our thesis we are going elaborate and see how to distinguish those proper nouns with clues that helps identify these types of nouns from an open text.

Verbs

Verb is the most vital part of a sentence that says something about the subject of a sentence, expresses an actions, events or states of being. Wolaytta verb classes occur mostly in the final positions of a sentence. Below the bolded, ones are categorized as verb class:

Chombey kaamiyaa **shamiis**/VB. (Chombe baught a car)

Cuuruqoy nu keetaa **yiis**/VB**. (**Cuuruqo came to our home**)**

Adverb

Adverbs are words that are used to modify a verb or an adjective or another adverb or a clause. They usually precede the verbs they modify or describe. An adverb indicates time, manner, place, cause, or degree and answers questions such as „how?“, „when?“, „where?“, and „how much?“. Below bolded ones denote wolaytta averbs:

Toomasi **eesuwan**/ADV yiis. (Tomas came **quickly**)

Birhanuy **zinno**/ADV Wolaytta gakkiis. (Birhanu arrived Wolaytta **yesterday**)

Adjective

An adjective is a word which tells us more about a noun. It also modifies a noun or a pronoun by describing, identifying, or quantifying words. In our case an adjective usually follows the noun or the pronoun which it modifies. Belowe bolded words denote wolaytta adjectives :

Abara zino ba **karretta/JJ** ishaara be'aas. (I saw Abera yesterday with his **black** brother)

Tana **addussa/JJ** asi lo'ees. (I like **tall** person)

Pronoun

Pronouns are used to make sentences less repetitive. Alike English Wolaytta **pronoun** can replace a noun or another pronoun. They are marked for number and gender. For instance, pronouns like "Ta/ tani" which means "me" masculine or feminine (singular), "'a" which means "she" is feminine (singular), "i" which means 'he' is masculine (singular), "etti" which means 'they' is plural and can be masculine or feminine and "nu/nuni" which means "we" is plural and can be masculine or feminine, some times "nu" denotes "our". Below highlighted denote Wolaytta pronouns.

Abebey kehaa asa. ***I*** daro too lo'o ossuwaan erettees. (Abebe is kind person. ***He*** is mostly known by his good work).

2.5.4 Named Entities in Wolaytta Language

It is perceived that NEs are words that are utilized to name an individual, association, location and different things. In this part, we will talk about the idea of NEs in Wolaytta. Our investigation zeroed in three classifications of NEs: PERSON, LOCATION and ORGANIZATION.

The nature and properties of NEs in Wolaytta generally takes after that of English. They are classified under the word classes of proper noun. In Wolaytta, NEs like name of an individual, organization and location share common qualities that, they are capitalized when written. Extra properties, for example, hint words and additions are additionally used to recognize NEs in Wolaytta.

Capitalization is the fundamental marker of NEs in which different properties depend on. It is one of those properties utilized in our study. Except whether they are of type numbers (like date, time, money related values), NEs start with a capital letter whether or not they are situated toward the start, center or end of the sentence as demonstrated underneath.

Manttaa Markossa Temesgeni Arbaminchee biis.

(Mr. Temesgen Markos went to Arbaminch)

Addis Ababa Unversittey tammareta ekkiidi de'ees.

(Addis Ababa University is in taking students)

Clue words are other helpful properties in distinguishing NEs in Wolaytta. Clue words are normal words that go before and after NE words, those words fundamentally work in combination with capitalization. In contrast to that of English, in Wolaytta, clue words for location and organization succeeds NEs they address, whereas person clue words precede the NE they represent. Below are examples clue words boldly highlighted:

Example of Person clue words:

Mantta Tomsasi naa'u nata aawa. (***Mr***. Tomas is father of two childs)

Professoree Tadassa Takeli Wolaita Sodo Universtee presdantte sunttetida.

(***Proffoser*** Takele Tadesse nominated as Wolaita Sodo Unversity president)

Mishirro Almaza giyaa bassu. (***Ms***. Almaz went to market)

Piresdaantte Zawude Sahleworka Ugandaa bitta gelassu.

(***Presedent*** Sahilework Zewude arrived in Ugandaa)

Example of Location clue word:

Wolaytta Sodo ***ambaan*** daro dichcha ossoy polettiis. (Much developments tasks were carriedout in Wolaytta Sodo)

Wolaytta ***mootan*** dumma dumma zanaqaa sohoti de'oosona. (In Wolaytta there are different tourism places)

Example of Organization clue word:

Arbaminchee zal'iyaanne unduustirre ***xifate ketta*** gadawatura ya'aay polettiis. (Arbaminch Trade and Industry Office held meeting with stakeholders).

2.5.5. Challenges of Wolaytta NER

One clear test experienced in attempting to tackle the issue of NER is that proper nouns like person names and organization name. The main ambigiuty happens when a similar name refers to different entities of a similar kind one for person and another for organization or location too. For instance, Kawo Tona branch is an organization name of commercial bank of Ethiopia, while Kawo Tona also can refer to person entity to handle those problem we used contextual chunkers. As (Belay, 2014) stated these difficulties are difficulties experienced by both morphologically poor and rich languages.

2.6 Summary

Reviewing over different priorly carried out articles, research works and paper in proceedings on particular thematic area is the vital and basic task for any research work on that particular focus area. Therefore, in this chapter we have gone through various literatures in order to understand deeply what the problem domain looks like. Hence, we have reviewed different literatures to attain current Wolaytta NER model. In related work section, various NER model developed globally and locally were articulated on our work. Finally, we have discussed about the current domain area that is Wolaytta Language. Here the nature, writing system, the way named entity were represented, challenges of WNER and word categories of the language were elaborated.

CHAPTER THREE

3. METHODOLOGY

This section portrays the methods and procedures used to achieve the objective of current study. We have described research design, method and approach followed in the study. The section also discusses tools (hardware, software and deployment), data sources, data annotation and preprocessing techniques used. At last, we enlisted the feature sets used for the model and the performance evaluation metrics used to evaluate the WNER model are discussed.

3.1 Design

According to Gizatie et al., there are four commonly used research approaches for any research studies, which are quantitative, qualitative, design oriented and mixed approach (Gizatie Desalegn., 2020). The quantitative research approach uses standardized measures, numerical values and analyze data using statistical technique. It can be applicable in phenomena that can be expressed in terms of numbers (Kothari, 2008). However, qualitative research approach focuses on exploring phenomenon, problem, or event; tends to be more in-depth and have smaller sample size (NJ, 1998). The mixed research approach is a procedure to analyze, collect and mix research of quantitative and qualitative methods in a single study (Uma D and Pansiri, 2011).

In this study, quantitative research approach in experimental method have used. Therefore, we have led various experiments to investigate the performance of selected machine-learning models. Moreover, performance measurement metrics (Precision, Recall and F1-score) we used are determined numerically.

3.2Method

To attain the objective of the current study we have followed various research methods. After identifying the current research thematic area we have gone through priorly conducted research works on the domain. However, we have found some limitations at a time of literature survey specifically from related works with our case. To overcome such limitations we have formulated research questions. The data was gathered from four selected sources Wolaytta Language and Literature Department, Wolaytta Wogeta Radio, Wolaytta FBC Radio Station and other researches. Then the preparation steps were took place to attain objective of the study.

Finally, designed proposed solution was implemented and results found were discussed. Based on these steps, methods followed to address the current problem are depicted (Figure 3.1).

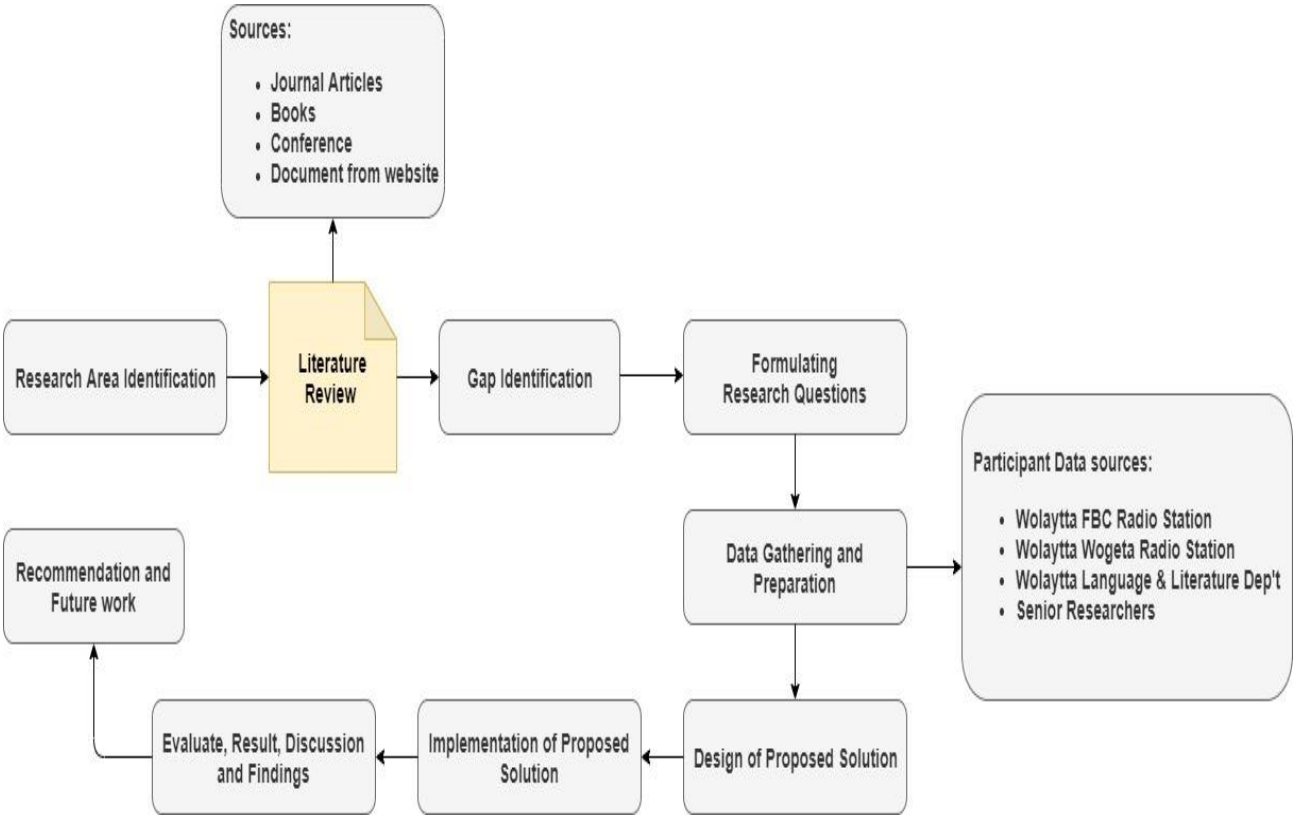


Figure 3. 1 Research Design of WNER

3.3 Approach Followed

The approach used in this research is Machine Learning approach. Machine learning focuses on the use of algorithms that can access features of annotated training data and use it to learn for themselves. We have preferred machine learning approach rather than other approaches(dictionary-based, rule based and hybrid) because it is economical in terms of time and human labor, also adaptability with new terms might make the approach suitable for NER model.

3.4 Data Collection

As data is the main pivot for current NER model, we have gone through four data collection sources. The sources we used are secondary data sources. We have discussed below our target data sources and their selection criteria.

3.4.1 Data Source Selection

The data collection sources were chosen after considering their content significance. These sources are Wolaytta Wogeta Radio Station (WWRS), Wolaytta Fana Broadcasting Corporation Radio Station (WFBCRS), Wolaytta Language and Literature Department (WLLD) and prior researches. In addition, we have also used prior researchers in the thematic area. From above data sources the sentences that comprise at least one NE from three (Person, Location and Organization) were gathered for the current study.

3.3.1.1 Wolaytta Mass Medias

More abundantly, the current study used data gathered from Social Mass Medias from the area. These are mass medias are WWRS and WFBCRS. We have selected Social Mass Medias as they work with Named Entities on their everyday activities. Therefore, from these sources 647 sentences were gathered with 9430 and 4530 words from each.

3.3.1.2 Wolaytta Language and Literature Department (WLLD)

We have selected WLLD, as they are more concerned with the language's morphological issues. Furtherly, they are enriched with digital data that was suited for the current study. From WLLD we have gathered 160 sentences that comprise 1624 words.

3.3.1.3 Prior Researches

The prior researchers in the language with different thematic areas in NLP are also targeted source of the study. However, from these data source we have gathered 100 sentences that consists 836 words.

Below we have elaborated gathered data with their percentiles in pie chart:

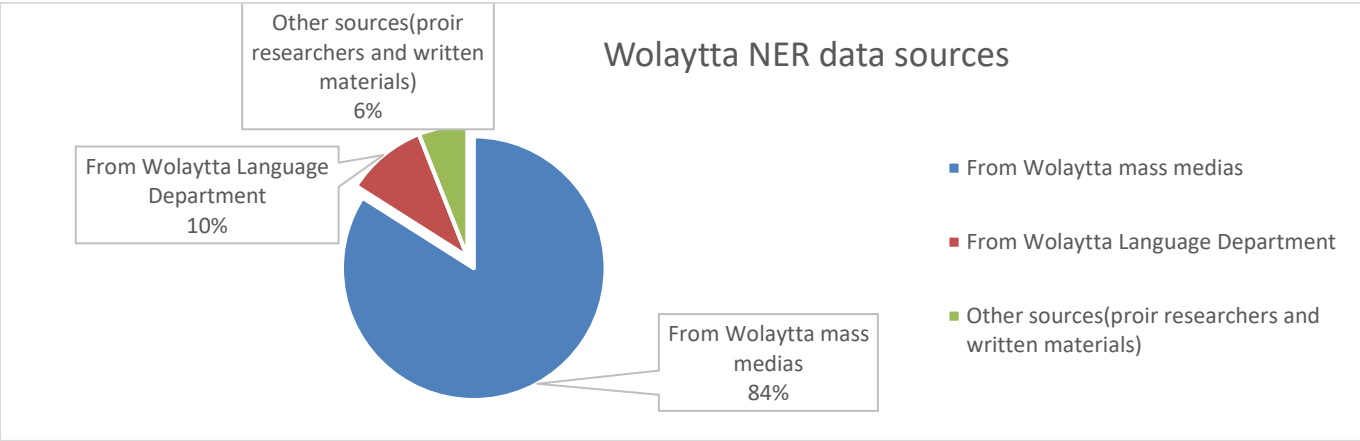


Figure 3. 2 Wolaytta NER data sources

Below is we have descried data sources with their statistics in tabular form :

Table 3. 1 Collected Data description

Data Sources	**Type of the source**	**Number of sentences collected**	**Tags Distribution**				**Number of Tokens**
			PER	**LOC**	**ORG**	**Other**	
WWRS	Mass media	405	632	582	601	7615	9430
WFBCRS	Mass media	242	432	472	419	3207	4530
WLLD	Educational	160	281	310	319	714	1624
Researches	prior researches	100	192	195	190	259	836
Total		**907**	**1537**	**1559**	**1529**	**11,795**	**16,420**

3.5 Dataset Preparation for WNER

This phase is the most troublesome and tedious task in current study since the performance of the model is exceptionally reliant on the nature of the dataset. Selected ML algorithms require dataset to be designed in a quite certain manner; hence, the gathered data have to pass though some preparation steps to help our model in yielding helpful insights.

As demonstrated in the beneath crude ML development pipeline, data preparation goes before training and feature extraction of any NER model.

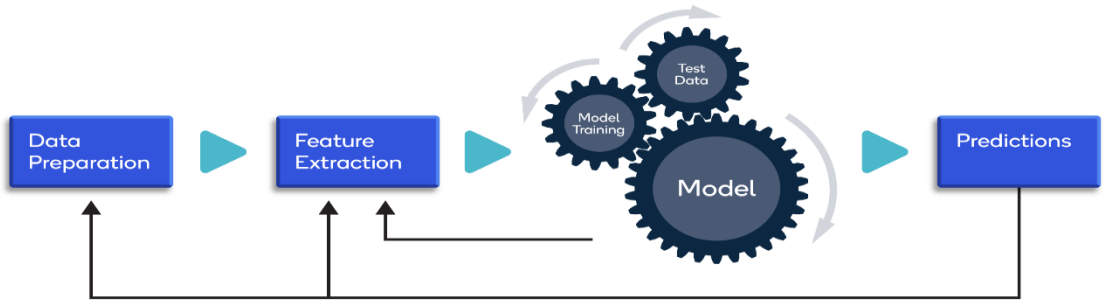


Figure 3. 3 Sample sequence of steps in NER model[6]

So, the gathered data have gone through data cleaning and filtering phases to mitigate the data quality issue. After cleaning and filtering our dataset have structured in one file.

Some of data handling procedures the WNER dataset passed were:

- **Inaccurate Data:** Unrelated data to NER problem could produce garbage output; hence, we have omitted unrelated data.
- **Missing data:** Sub-data with missing column values were handled from the dataset.

[6] https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk/learning-resources/

3.6 Dataset Annotation

The data that passed preparation stage was given to human annotators. Manual data annotation was followed in the current study. Since, existing annotator software tools were incompatible with our local language we use manual technique. As a result, we have used four domain experts for the dataset annotation task. The annotation was carried out by following annotation Guideline in Appendix B. Our annotators were chosen from WLLD in Wolaytta Sodo University after assuring their educational background on linguistic aspects of the language.

For a sake of reliability, more than one annotator is required and consideration have taken to annotator agreement. In section 3.6.1, we have discussed how annotators can agree if their tagged NE classes differ. After successful completion of annotation, the validity of annotated dataset have approved by WLLD as can be seen in Appendix B.

3.6.1 Inter-Annotator Agreement

Obviously, Named Entities have overlapping behavior. In our study, the same entity name could belong to more than one NE class. In a sense, there may be organization name entities at the same time that refer to person name. For instance, as can be seen from below phrase:

"*Tophphe Zal"iya Bankiyaan Kawo Toona Daashshaa*" (literally, *In Commercial Bank of Ethiopia Kawo Toona Branch*) in this phrase "Kawo Toona" is PERSON entity at the same time "Kawo Toona Daashshaa" is bank ORGANIZATION name in Wolaytta.

As every annotator has its own contextual understanding to label the data, such phrases may bring annotator bias to the dataset, hence that may have great impact on the model. According to Mor et al., perfect annotator agreement can improve model performance (Mor Geva, 2019). The current study utilized Cohen's kappa evaluation for inter annotator agreement.

3.5.1.1 Cohen's Kappa

Cohen's Kappa is a standard statistic tool that measures inter-annotator agreement. This kappa statistic, which is a number between -1 and 1. The higher value means complete agreement; zero or lower means chance agreement of the annotators. It is known by expressing the level of agreement between annotators on a classification problem.

It can be defined as:

$$\kappa=(Po-Pe)/(1-Pe)$$ ***Equation 3. 1***

In the above formulae, *Po* is the observed probability of agreement between annotators on the label assigned to any sample, and *Pe* is the hypothetical probability of chance agreement when both annotators assign labels randomly. It is preferred if the agreement value became 0.8 and above. Detail annotation result was described in Chapter 6 (section 6.3).

Agreement interpretation	Kappa values
Less than chance agreement	Less than 0.0
Slight agreement	Between 0.0 and 0.2
Fair agreement	Between 0.2 and 0.4
Moderate agreement	Between 0.4 and 0.6
Good agreement	Between 0.6 and 0.8
Very Good agreement	Between 0.8 and 1.0
Perfect agreement	1.0

Table 3. 2 kappa's Inter Annotator agreement values standards

3.5.2 Annotation Format

Named Entity recognition annotation data format comes in two ways either a Pandas Data Frame or a path to a text file containing the data[7]. Pandas Data Frame annotation data format was used in current study. We have followed IOB encoding scheme of shared task of CoNLL-2002[8] to tag WNER dataset.

The IOB tagging scheme was suited to identify the Beginning and Inside of certain entity. While tagging B-X denotes the beginning of an entity X and I-X denotes Inside of entity type X. Hence,

[7] https://simpletransformers.ai/docs/ner-data-formats/
[8] https://www.clips.uantwerpen.be/conll2002/

we tagged our data as B-PER, I-PER for person entity, B-LOC, I-LOC for location entity, B-ORG, I-ORG for particular organization entity and O (other) used to denote other non-named entities.

As stated in section 3.4.1, sentences that contain at least one NE are used for the dataset annotation. Then in the chosen sentences word level annotation was carried out. Each word of the sentence are labeled with their respective POS tags and named entity tags. Then final annotated dataset was saved in structured tabular format with .CSV file extension. For further reference, see sample annotated dataset snippet in Appendix B.

3.5.2.1 Named Entity Tagset

In our current work, the NE tagset used have further subdivided into detailed categories to denote the boundary of NEs properly. We have seven NE tagset used in the dataset. These are namely, B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG and O label for denoting non-NE tags.

3.5.2.2 Legal Tag sequence

As we have followed BIO tagging scheme, there is a sequence rule NE tagged tokens must follow. The legal tag sequence is described on Table 3.2 below where the first column lists the representation of the tags and the second column lists the tags that may succeeding them, with the variables ranging over all NE types.

Table 3. 3 BIO legal tag sequence

Tag	**Legal Following Tags**
O	O,B-X
B-X	O,I-X,B-Y
I-X	O, I-X, B-Y

From the above table we can conclude that during the legal tagging of the BIO scheme *Begin (B)* and *Inside(I)* tags have the same legal followers. Whereas, Other (O) tag must have to come with O and B-X succeeding tag sequences. B-Y denotes to the first token of another NE type.

3.5.2.3 Final Annotated Dataset Description

Here we have attempted to describe the distribution of entity classes throughout the dataset. As the problem on hand was classification problem, the dataset have also annotated on the same manner. Hence, to attain the classification goal we ensured dataset on hand is clean, relevant and understandable by selected ML algorithms. Below the distribution of labeled entity classes was depicted, most of the dataset was comprised with Outside (O) tag. This happened as each sentence in the dataset can have various non-named entity words. Therefore 'O' tag represents roughly 70% of all dataset in the current study.

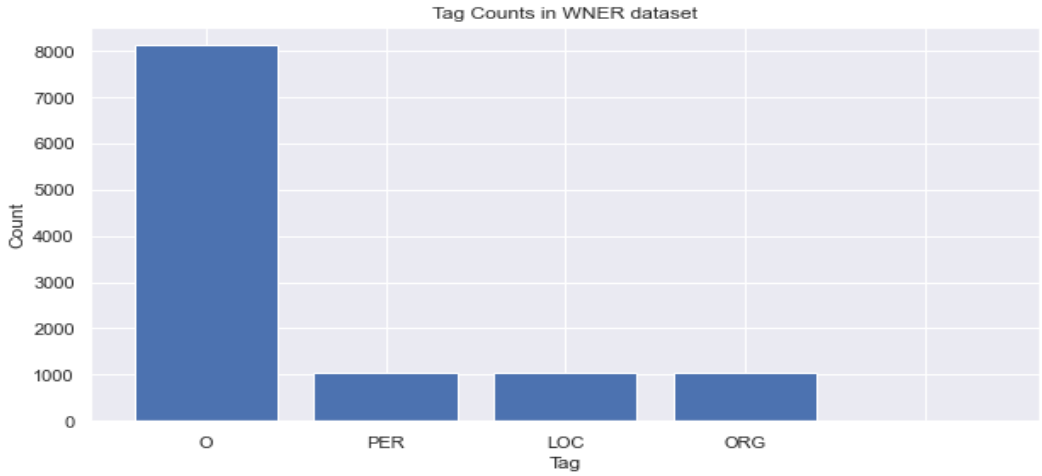


Figure 3. 4 WNER Tag counts including "Other" non-named entities

Let us examine the same distribution without the 'O' label (by excluding non-named entities):

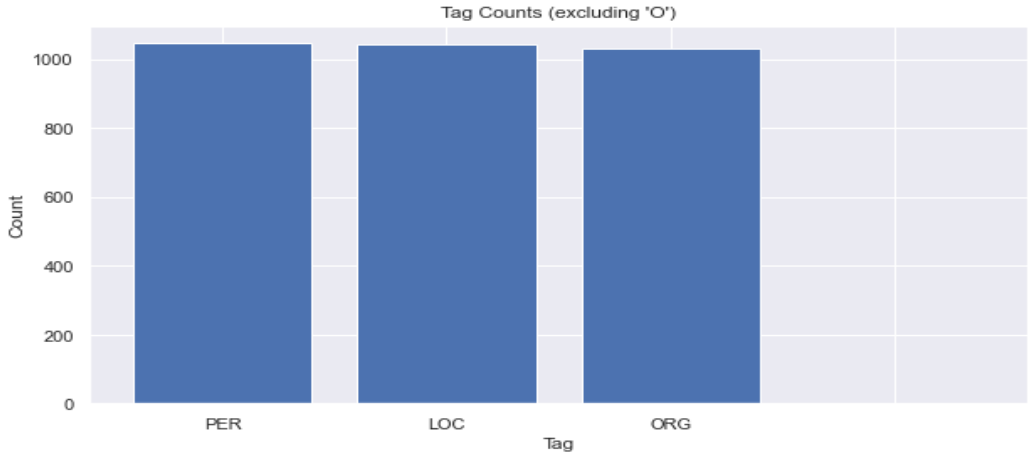


Figure 3. 5 Tag Frequency excluding 'O'(Other non-named entity) tags

Above from *Figure 3.5*, beside the exclusion of "O" tag we can visualize that in the annotated WNER dataset the distribution of three tags namely Person, Location and Organization were balanced. Their balancedness could be important in model building, because their features could not dominate each other. Though, the total annotated dataset size for this study is 16,420 from these named entities comprise 4625. Those three classes Location, Person and Organization sized 1559, 1537 and 1529 respectively.

3.5.3 POS Tagging

As research works of (Belay, 2014) and (Ahmed, 2010) clarified POS tag of particular token plays crucial role in identifying NE's on their respective task. Even though POS tagger for Wolaytta were developed priorly by (Herano, 2015) and (Fikre, 2020) but in their work NE's were rarely occurred, since made their work incompatible with current study. In addition, proper noun class that was backbone for our work did not tagged clearly, they simply included proper nouns as generic noun class.

After considering the above cases, Part-of-speech tag of words was used as an internal feature in Wolaytta NER. We incorporated Part-of-Speech labeling, as it not only assign a token to the exact syntactic category, also gives inflectional categories of the NE classes person, location, organization.

3.7 Wolaytta NER Modeling

3.7.1 Preprocessing

As data preprocessing is the first and crucial step while creating a machine learning model, after putting our data in a formatted way we went through preprocessing. In this phase low-level formatting, cleaning, tokenization, stop word removal and normalization took place.

3.7.1.1 Low-level formatting

In this phase we have gone though elimination of irrelevant contents (junk files) that appear on our collected corpus. Some of these junk files that might be irrelevant for us were contents like Pictures, Tables, Diagrams and Advertisements are cleaned manually from our dataset.

3.7.1.2 Cleaning

Even though, NER model is deployed often in raw dataset we have gone through some cleaning processes to assure quality of the dataset. Hence, after low-level formatting cleaning took place that might remove all irrelevant contents like non-wolaytta strings. Finally the dataset left with Named Entity populated sentences. Thus NE's were three in our case as depicted in below sample word clouds.

(a) PERSON entities (b) ORGANIZATION entities (c) LOCATION entities

Figure 3. 6 Sample Named Entity (highlighted in bold) Word clouds in Wolaytta Corpus

3.7.1.3 Tokenization

Tokenization is crucial preprocessing phase in our work. It is responsible for splitting the given input sentence in to words (tokens). The current dataset was prepared with token level, so it made easy task to identify token boundary.

3.7.1.4 Normalization

Normalization of entity types is crucial for comprehensive document querying in NER model (Weston, 2019). Wolaytta language was characterized by rich inflectional morphology (WAKASA, 2008). So, words with different spelling but have common meaning were normalized in our dataset. For instance, some of our data gathering sources miss spelt word "Wolaytta" as "Wolaita" such words were came to common and grammatically correct order.

3.7.2 Feature Extraction Techniques

In textual corpus feature extraction is the technique of transforming a list of words in the dataset into a feature vector that is preferable by machine learning models. Therefore, dataset that have passed preprocessing techniques was represented in feature vector form for algorithm consumption. By Considering Wolaytta language's morphological structure, we have made list of features that ought to be effective in solving named entity classification task. The selected models cannot be directly trained on a corpus annotated with named entities (Jing Li, 2018). The corpus needs to be reworked into a group of representative feature instances. After considering the above reasons, we have used Word-Level Features with DictVectorizer encoding technique to make current dataset compatible with selected algorithms.

3.7.2.1 Word-Level Features

As we have stated in section 2.3.5, these features pertain to the characters that make up a word as well as its grammatical sense. They attempt to check every tokens shape including contextual information. The current study used context size of three. Moreover, currently employed word level NER features are categorized in two main categories, language independent and language dependent. Below we attempted to list of all the features that were used in current study.

Language independent (basic and orthographic) features used:

- ✓ *w* – The original token.
  - ✓ *Lowercase (w)* – w in lower case letters (e.g. Lowercase (Sodo) = sodo). The usage of this feature instance decreases the amount of tokens that can be treated differently because of their case (especially relevant for common words that come at the beginning of the sentence).
  - ✓ *Suffix-n (w)* – suffix that determines the last n (in our experiments $2 \leq n \leq 5$) characters of w in lower case (e.g. Suffix-5 (Toomasinne) = sinne; Suffix-5(Markoosinne) = sinne). This feature should help to extract inflectional Wolaytta endings. E.g., the roots in "Gakkanaas" and "Ottanaas" are different, but they share the same ending in locative case "as".
  - ✓ *IsUpper (w)* – Boolean indicator that determines if the current word *w* is all capitalized (e.g. *IsUpper* (AMBAA) = true).

- ✓ *IsDigit (w)* – Boolean indicator that determines if the current string is numeric (e.g. *IsDigit* (201) = true).
- ✓ *IsAlpha (w)* – Boolean indicator that determines if the current string is all letter (e.g. *IsAlpha* (tamasgaana) = true).
- ✓ *IsAlphaNumeric (w)* – Boolean indicator that determines if the current word is mixture of numbers and letter (e.g. *IsAlphaNumeric* (2tto) = true).
- ✓ *havePunctuation(w)*- Boolean indicator that ensures if the current word have punctuation inside (e.g. *havePunctuation* (Naa"a) = true).
- ✓ *IsTitle (w)* – Boolean indicator that determines if the first character of w is capitalized (e.g. *IsTitle*(Ambaa) = true).

- ✓ Language dependent features used:
  - ✓ *POS (w)* – the part-of-speech tag of w (e.g. POS (Yohanissaa) = noun).
  - ✓ *POS (w+1)* – the part-of-speech tag of next word
  - ✓ *POS (w-1)* – the part-of-speech tag of previous word

We will carry out sequence of experiments with above language dependent and language independent features to find out most suitable combination of features sets for NE tagging of Wolaytta. The detail explanation is illustrated in Chapter 6. In order to feed our model we have given the above word level features to DictVectorizer. DictVectorizer converts the whole dataset in to vector form that is understandable by classifiers.

3.7.3 Modeling

After the loaded dataset was preprocessed and represented in feature vectors, it was fed in to our classification machine learning models. We will use three different machine-learning algorithms Conditional Random Field, Support Vector Machine and Random Forest. From those best-performed model would be chosen as Wolaytta NER model. As the current work was its first attempt, choosing optimal algorithm for the problem is a comprehensive task that demands the analysis of a variety of factors. We will choose the algorithm by evaluating tasks like accuracy, interpretability, complexity and scalability with our problem domain.

3.7.4 Modeling Criteria

We have chosen the above algorithm because of their current state-of-art behavior. Below we have discussed why we have preferred to use them:

3.7.4.1 Conditional Random Field

Here we will use linear chain CRFs for the Named Entity model. It is statistical modeling method often applied in machine learning and used for structured prediction. We have chosen it because, as it takes context (neighbors of particular word) into account at the time of prediction. As work of (Alemu, 2013) stated efficiency of dealing with non-independent, overlapping features and their ability of overcoming the label bias problem made CRF algorithm most suited for NER. In addition, they are well known conditional model without label bias problem.

Therefore, the conditional probabilistic sequential and undirected graphical behavior of CRF model made it current state-of-art for NER model. Hence, undirected behavior allows the model to consider surrounding contextual information before predicting. Below we have depicted with sample variables in which undirected graphical model globally conditioned on variable *X*.

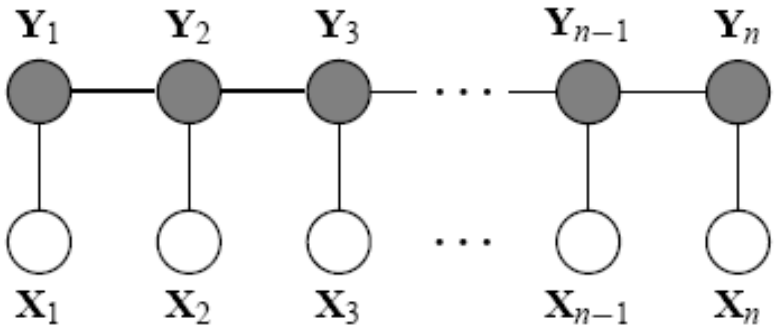


Figure 3. 7 CRF Graph Model

Given an undirected graph $G = (V, E)$ such that $Y = \{Yv \mid v \in V\}$, if

$$p(Y_v \mid X, Y_u, u \neq v, \{u,v\} \in V) \Leftrightarrow p(Y_v \mid X, Y_u, (u,v) \in E) \qquad \textit{Equation 3. 2}$$

the probability of vertex Y_v given X and the random variables corresponding to the nodes neighboring v in G. Then (X, Y) is a conditional random field (Sarkar, 2020).

For more clarity, CRFs model works based on probability p(y|x) of a possible output sequence y = (y_1….y_n) ε Y given the input sequence x = (x_1….x_n) ε X which is also said to be the observation. Here Y is the set of all possible output tag sequences and X is the set of all possible sequences of observations. As we stated before our linear chain CRFs is a special form of current CRFs for problem at hand, which is structured as a linear chain that models the output variables as a sequence.

Let consider below formulation;

Equation 3. 3

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i)\right)$$

Where *j* indicates the length of the sentence, *k* indicates the number of feature templates, $\boldsymbol{\lambda j}$ represent the weights assigned to the different features in the training phase and $Z\lambda$**(**x**)** is a normalization factor variable that make the probability in the limit between [0,1]

In our linear-chain CRFs there is a technique known as feature functions that are the key components in current research work. Hence, instance feature sets are extracted by feature function method. For simplifying description, the general form of feature extraction is $t_j(y_{i-1}, y_i, x, i)$, which looks at a pair of adjacent states y_{i-1}, y_i, the whole input sequence x, and where we are in the sequence (i).The above arbitrary function outputs real value in our case.

For instance, let us define a simple feature function which produces binary values as,

$$t_1(y_{i-1}, y_i, x, i) = \begin{cases} 1 & \text{if yi=PERSON and xi=Tadassa} \\ 0 & \text{otherwise} \end{cases}$$

The above equation means if the i^{th} word is Tadassa and having a tag PERSON, then t_1 is one otherwise t_1 is zero. The importance of this feature relies on its corresponding weight $\boldsymbol{\lambda_1}$. If $\boldsymbol{\lambda_1} > 0$, whenever t_1 is active (i.e. we see the word Tadassa in the sentence and we assign its tag PERSON), it increases the probability of the tag sequence $y_{1:n}$ (y).This is another way of saying the CRFs model should prefer the tag PERSON for the word Tadassa. If, $\lambda 1 < 0$, the CRF model will try to

avoid the tag PERSON for Tadassa. In this case the value of λ_1 will be learned from the WNER dataset.

Other instance,

$$t_2(y_{i-1}, y_i, x, i) = \begin{cases} 1 \text{ if } yi=\text{PERSON and } x_{i-1}=\text{Mantta} \\ 0 \text{ otherwise} \end{cases}$$

This feature becomes is active if the current tag is PERSON and the previous word is "Mantta". So, we would expect a positive λ_2 to go with the feature. Here we can note that both t_1 and t_2 can be active for sentences like "Mantta Tadassa" at different *i* values. This example of overlapping features boots up the belief of y_2 = PERSON to $\lambda_1 + \lambda_2$. The same thing is true for other NE classes. This is why we have chosen CRF in our study since it support overlapping feature sets that other ML classifiers lacked.

3.7.4.2 Random Forest

As stated by Louppe et al., Random Forest is a supervised classification ML algorithm. The ability to limit overfitting without substantially increasing error due to bias is the reason we have chosen it, thus they are such powerful models. Also work of (Breiman, 2001) and (Louppe, 2015) stated, random forest's feature of reducing variance by training on different samples of the data made it state-of-art model.

3.7.4.3 Support Vector Machine

The current Named Entity Recognition problem is sequential problem. SVM is well known linear model often used in classification problems (Thoudam D, 2009). Although we used the model for this study to solve the sequential behavior of current NER task. The algorithm plots each data item as a point in n-dimensional space with the value of each feature instance being the value of a particular coordinate. Then, categorizing of different classes would be undertaken by finding the hyper-plane that distinguishes the four classes very well in our study.

3.7.5 Model Evaluation Metrics

In machine learning model evaluation is most mandated task to assess the goodness of fit between model and dataset. It measures the performance of how well the model works on unseen data. Also, more significant to evaluate the generalization ability of the developed model on new data.

According to Nigatu et al., there are two approaches to evaluate the given model performance in text classification problem. The first was holdout approach and the other one was k-Fold Cross Validation approach (Nigatu, 2021).

In holdout approach the model is evaluated based on data that was out sampled (not seen) by the model during training. This approach is characterized by simplicity, speed and flexibility, but has drawback in resulting differences in estimated accuracy because of its high variability among the training, validation, and test sets. While k-Fold Cross Validation approach is currently widely used. It is known for its skill in the less biased estimation than the previous simple train/split approach. This approach also abundantly reduces variance, because of above reasons k-Fold Cross Validation is preferred model evaluation technique in current study.

3.7.5.1 K-Fold Cross Validation

In k-Fold Cross Validation technique dataset resampling procedure is used to evaluate machine-learning models on a limited data sample (Nigatu, 2021). The technique have one known parameter named as *k* that implies to the number of folds that a given data sample is to be split into. Hence, in the current study we have applied 5- Fold Cross Validation.

The way in which k-Fold Cross-validation approach works is:

1. Split the whole dataset randomly into *k* folds. Here the value of *k* should not be too high or too small. Therefore, ideally we have to choose 5 to 10 k values depending on the dataset size.
2. Then, fit the model by using the *k*-1 groups or folds and validate the model using the remaining k^{th} fold.

3.6.5 .2 Confusion Matrix

Confusion matrix is a tabular description that depicts the correctness and accuracy of the model. As the current study is carried out through supervised ML approach, the confusion (error) matrix would become capable to allow visualization of performance of selected algorithms. It will show the distribution of predicted true values.

3.7.5.2 WNER Classification Metrics

As we have stated in literature review (section 2.3.6) when you train a particular NER system the most typically used evaluation metrics were to measure precision, recall and f1-score at a token level.

Therefore, in current study the evaluation was performed by comparing the system output to the human-prepared dataset in terms of the precision (P), recall (R) and their harmonic mean, the F1-score.

Precision

In the easy words, precision is the ratio between the True Positives and all the Positives. In our study case, ratio between number of NEs correctly found by the model and number of NEs classified as NE by our model.

Mathematically:

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

Recall

Also, recall is the measurement value of how our NER model correctly identifies True Positives. That means, recall tells us from all the tokens that actually are NEs, how many are correctly identified as NEs by our model. Mathematically see below:

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

F1-Score

F1-score is the harmonic mean of the precision and recall.

$$F1 - Score = \frac{2PR}{P + R}$$

We have described terminologies used in our evaluation metrics:

True Positives (TP): These are cases where tokens are Entity type-X and their predicted class is Entities type-X. Where X is one of our four entity classes.

True Negatives (TN): When our model predicted tokens as out of Entity type-X and their actual class is out of Entity type-X.

False Positives (FP): When our model predict a token as Entity type-X, but they do not belong to Entity type-X.

False Negatives (FN): When our model predict a token as out of Entity type-X, but they actually belong to Entity type-X.

3.8 Tools

Here we discussed tools (may come in three forms hardware software and deployment) that took place on our NER model building process that are used in data preparation, model training and implementation.

3.8.1 Software

There are many tools used so far by researchers in the domain area such as work of (Ahmed, 2010) and (DAWED, 2020) used Ling pipe for their NER development. Also Mikiyas (Belay, 2014) and Dagimawi (Demissie, 2017) used WEKA with java implementation and WEKA with Apache openNLP respectively. This study used the python programing language for implementing and experimenting proposed solution. Table 3.4 and Table 3.5 shows the list of software tools and python packages with their version and description.

Table 3. 4 Software tools used

Tools	**Version**	**Description**
Anaconda navigator	3.7.0	It is a python desktop graphical user interface used to provide conda packages, environments and channels.
Jupyter Notebook	3.7.4	It is python web application used to create live code, visualization, and narrative text.
Spyder (Anaconda)	3.7.4	The Spyder (Anaconda) is also used to write python code.
Python	3.7	Interpreted, high-level and general-purpose programming language that is widely used in machine learning to harvest insights from particular dataset.
Microsoft Excel	2016	Used data preparation tasks in cleaning filtering and sorting the data and used for annotation
Microsoft Word	2016	Used to write the documents
Draw.io		Online drawing tool used to draw diagrams, architectures, UML and soon

Table 3. 5 List of Python Packages used

Python libraries used	Description
Scikit-learn	It allows us to work with classification, regression and clustering algorithms
ELI5	Eli5 is a python package that helps to debug machine-learning classifiers and explain their predictions.
NLTK	The platform used to work with human language data.
Pandas	Used to read, write and manipulate data frames
Numpy	Used for process multi-dimensional array in python
Pickle	Used to store and convert python object in database or file.
Matplotib	Used to visualize data in python
Seaborn	Used to visualize data in attractive interface in python
Regular Expression	The re module allows us to search a string for a match in python. In this study, it is used for data cleaning and data normalization.
String	Byte of array representing Unicode characters in python.

3.8.2 Deployment

Flask[9]

Finally, after performing various experiments well-performed model is deployed in Flask web server. Hence, the deployed model will serve users with real time NER task. We have preferred Flask as it provides WSGI that is interactive web application platform suited for users.

3.8.3 Hardware

Hardware tool we utilized to build the model and over all other tasks was a Personal Computer with an Intel® processor Core™ i5-5300U CPU @2.30GHz, 8 Gigabyte of actual memory, 1 Terabyte hard disk storage capacity, and 64 bits Windows 10 Pro operating system.

3.9 Summary

As methodology was overarching strategy and rationale of any research. In our methodology, procedures or techniques used to prepare corpus, to select effective model and how we are going to accomplish Wolaytta NER model was discussed systematically. Hence, here basically we have covered research design, method used, data collection issues, data preparation and annotation, data preprocessing tasks, model selection issues and finally depicted feature sets used in our NER model development.

[9] https://flask.palletsprojects.com/en/2.0.x/

CHAPTER FOUR

4. DESIGN OF WOLAYTTA NER

4.1 Overview

In this section, we discussed about the overall design of Wolaytta Named Entity Recognition model. At the beginning, we elaborated the glimpse of the importance for WNER design. The description of the WNER architecture is described along with flow of operations and subcomponents. Finally, we have described WNER deployment prototyping architecture.

4.2 Wolaytta Named Entity Recognition Architecture

According to Legesse et al., there was no common architecture for NER framework, because writing style used in certain languages, domain dependability of the NER task and approach followed might vary (Legesse, 2012).

Any NER model may comprise various modules with their respective sub-components to attain common objective at hand. Therefore, for the sake of clarity designing systematic flow of operations became mandate for the study. We attempted to depict generic architecture of certain NER model regardless of approach used in *Figure 2.1*. However, in this part we introduced a more explicit design used for WNER model. The below *Figure 4.1* is the general architecture we have employed for this study.

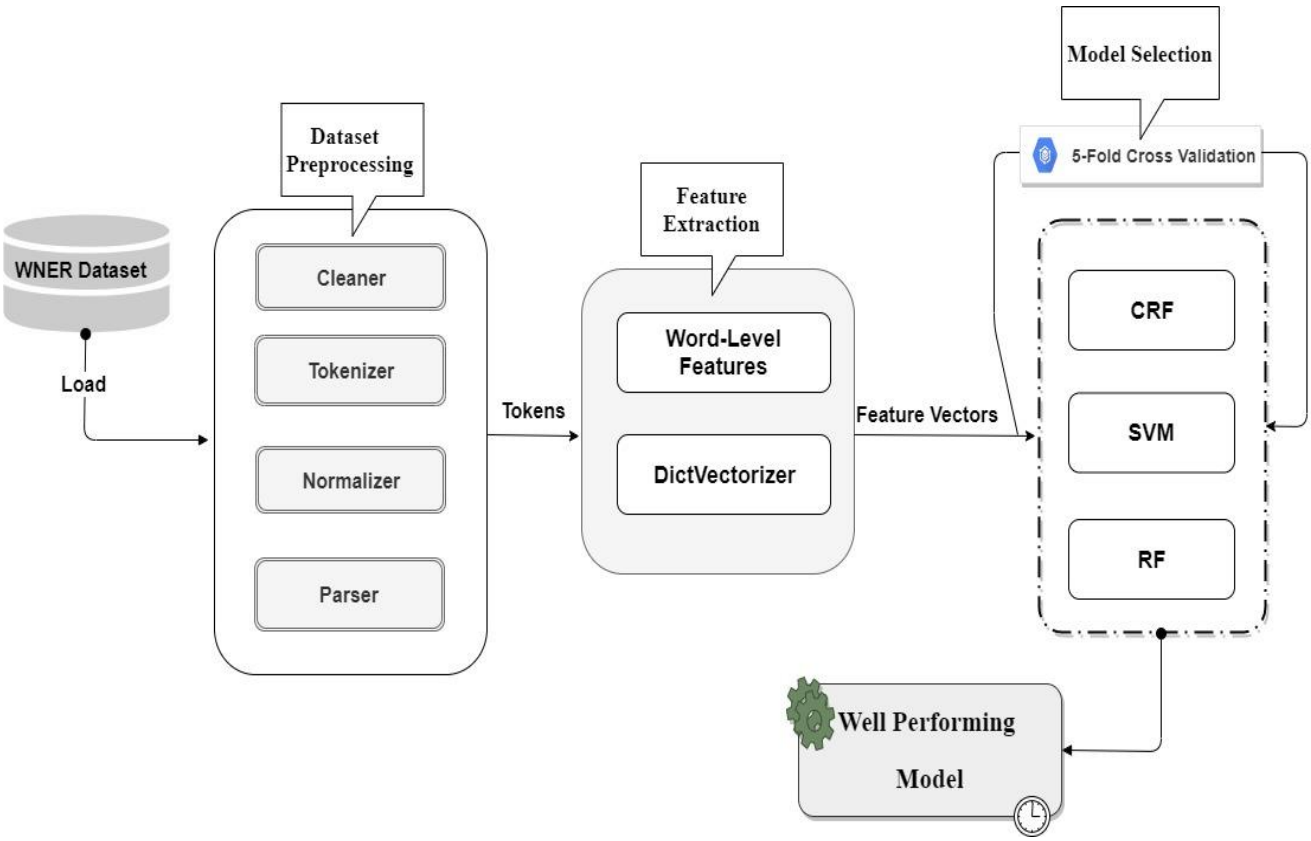


Figure 4. 1 Architecture of Wolaytta NER model

4.3 Phases of Wolaytta NER Architecture

Our architecture have mainly three phases; pre-processing phase, feature extraction and modeling phase. The pre-processing phase itself contains subcomponents like cleaning, stopword removal, normalization, tokenizer and parser. Feature extraction component also have two subcomponents word-level features and DictVectorizer for generating instance feature vectors of the dataset. Finally, model selection component comprises models used in this study and their evaluation technique employed.

4.3.1 Pre-processing Phase

This phase plays crucial role for every NER framework in building high performing NER model, by assuring the quality of data consumed by the ML algorithms. As a result, our pre-processing tasks took place on whole dataset. Those the phase have comprised Cleaning, normalizer, and Parser and Tokenizer modules in our design. Below we have described each modules:

4.3.1.1 Cleaning

This module of the architecture is responsible to eliminate irrelevant characters and emoji's if exist. Since, our dataset was collected from different primary sources and those sources have some irrelevant parts. However, not all punctuation marks are irrelevant in our case, for instance question mark, full stop and other punctuations admitted to be useful are not cleared because they might help us by determining end of certain sentence. Since, to ensure the model performance it became mandate to clean our dataset. Below is pseudocode to do so.

Algorithm 4. 1 *Algorithm for Cleaning WNER dataset*

Input: raw unprocessed WNER dataset

Output: processed clean WNER dataset

INIT:

Read WNER dataset

While (it is not the end of file):

If text contain special characters and symbols [')', '(', '-', '|', '$', '&', '/', '[', ']', '>','%', '=', '#', '*', '+', '\\', '•', '~', '@', '£', '_', '{', '}', '©', '^', '®', '`', '<', '→', '°', '€', '™', '›', '←', '×', '§"]

Then

Remove characters

If text contain emoji's [😀, 😊, 🙆, 😀…]

Then

Remove emoji's

If text contain nan(not a number)

Then

Drop nan and fillna

Return processed text

End if

Halt

4.3.1.2 Tokenization Module

Tokenizer module is the other pre-processing component in the architecture. It is responsible to read the Wolaytta plain text from sentence and performs tokenization. It is the process of breaking up a given sentence into list of tokens. The tokenization was performed by built in tokenizer component within python package.

4.3.1.3 Normalization

In Wolaytta language, there are many words with the same meaning but have different written forms. For instance “de’oi” would be normalized to “de’oyi”, “ekanau” normalized to “ekanawu” because in grammar of the language two consecutive vowels cannot came together. Since, these words must need to come in one grammatical form; hence doing this will reduce the confusion of our model with different word forms. The normalization component is mandated to do this task in WNER architecture. Below is sample algorithm that depicts how normalization module works.

Algorithm 4. 2 *Algorithm of dataset normalization*

Input: un normalized WNER dataset

Output: normalized WNER dataset

INIT:

1. Read WNER dataset

2. **WHILE** (it is not the end of file):

IF dataset contains [ai], **THEN**

Replace with ‘ayi’

IF dataset contains [au], **THEN**

Replace with ‘awu’

IF dataset contains [oi], **THEN**

Replace with ‘oyi’

IF dataset contains [Kataama], **THEN**

Replace with ‘Allaana’

IF dataset contains [Alamiyaa], **THEN**

Replace with ‘Salo Gufantuwaa’

ENDIF

3. **HALT**

4.3.1.5 Parser

Named Entity labeling took place manually by human annotators, which might make the task prone to errors. Therefore, parser is the essential component that checks the correctness of NE labeled training dataset and report if there are errors. Since, those errors may be tag inconsistency errors with the rules of CoNLL-2002 and mistyping errors. Such error may mislead other subsequent tasks. As a result, our parser module is responsible to check our dataset whether it is legally annotated or not. It checks legal tag sequence accordingly with *Table 3.3*. Below in *Fig.4.3* we have adopted pseudo code that shows how the parser works from research work of Ahmed et al.

```
For all lines in the WNER dataset
    Read a line
      If a word is not tagged
          Report error
      If the previous tag is O
          Check if the current tag is one of O & B-X
      Otherwise report error
      If the previous tag is B-X
         Check if the current tag is one of O, I-X, B-Y
      Otherwise report error
      If the previous tag is I-X
        Check if the current tag is one of O, I-X, B-Y
      Otherwise report error
End for
```

Figure 4. 2 Parsing algorithm for WNER model (Ahmed, 2010)

4.3.2 Feature Extraction Phase

As, choosing informative and discriminating features is a crucial step for effective algorithms in classification problem (Ahmed, 2010). In our case features are individual measurable properties of a text that increase the confidence level of predicting a given token as a NE, by providing necessary information associated to that token. Therefore, our feature extraction module identifies and extracts all the necessary features from the provided tokenized data. Then feeds extracted feature vectors to selected models. These whole tasks are carried out through two subcomponents word-level features and DictVectorizer.

4.3.3 Model Selection Phase

The current study have used three algorithms for WNER modeling. This phase enlists those models and their evaluation. The models we used were CRF, SVM and RF. We have evaluated the models with 5-fold cross validation. After undertaking evaluation well performed model was chosen as WNER modeling. Well performing model is the model that have better performance than the others do.

4.4 Wolaytta NER Prototyping

Our final trained and outperformed model is deployed in Flask server to help in users query. Flask is a web application framework that is written in Python. It is based on the Werkzeug Web Server Gateway Interface (WSGI) toolkit, which is provides a universal interface between the web server and the web user applications. Therefore, we have pickled our model in ".sav" file format, later on the model will predict on input from users by loading saved model from Flask webserver through WSGI platform.

Here, NE Recognizer component is responsible for recognizing user input. It takes input plain text from a user, recognizes NEs out of it and supplies NE recognized Wolaytta text as an output. Below we have elaborated graphically from right to left WNER prototyping model.

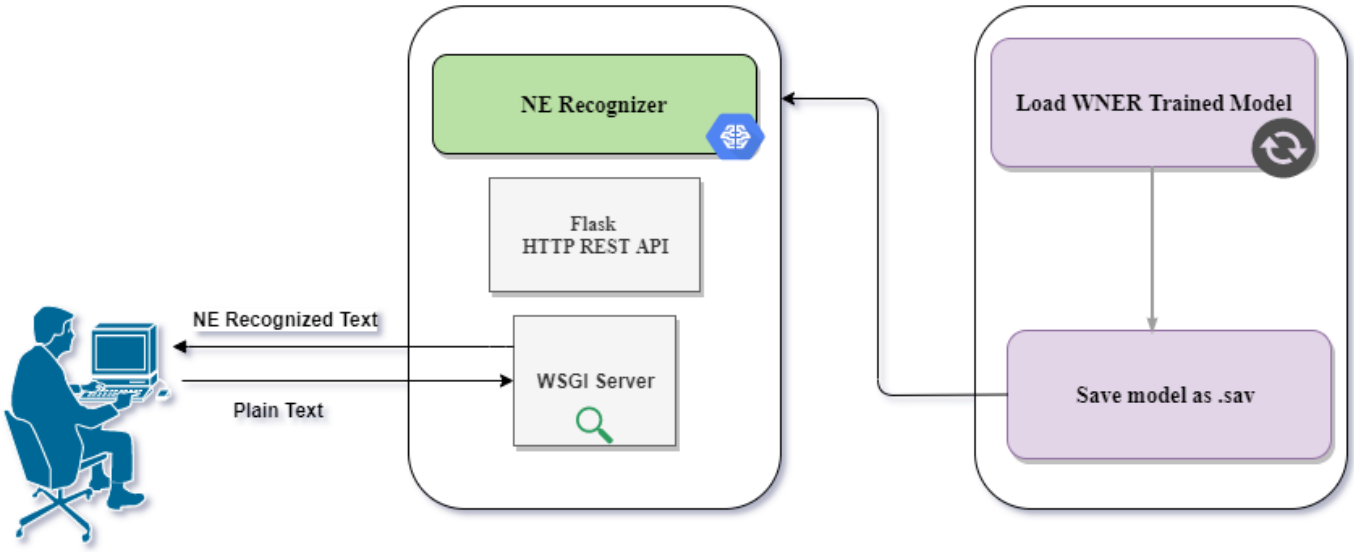


Figure 4. 3 Wolaytta Named Entity Recognition Deployment architecture

4.5 Summary

In whole sense, this chapter dealt with architectures used in our study. The WNER architecture above depicted can help us to understand work flows from one process to another including their sub modules. The second architecture depicted is prototyping architecture that shows us how user input can be processed and generated as NE recognized text from the developed model.

CHAPTER FIVE

5. IMPLEMENTATION OF WOLAYTTA NER MODEL

5.1 Overview

In this section, we have elaborated implementation of proposed study. The section addresses implementations of loading dataset, preprocessing, feature extraction and selected model implementation. Generally, it is about visual description of experimentations we undertook.

5.2 Implementation of Preprocessing

5.2.1 Importing Libraries

Importing python library packages is the foremost step in current study implementation. It is obvious that importing required library is mandate to use that particular library throughout our program. Hence, we have imported below libraries for our WNER model building.

```
#Importing required libraries for WNER
import numpy as np
import pandas as pd
import seaborn as sns
import Flask
import re
import nltk
import pickle
import matplotlib as plt
import numpy as np
import requests
from nltk.corpus import stopwords
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn_crfsuite import CRF
from sklearn.model_selection import cross_val_predict
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer,TfidfVectorizer
```

Figure 5. 1 Importing different libraries

5.2.2 Loading Dataset

Before doing any advanced preprocessing task, dataset should be made available for our python libraries. We used Pandas software package to load our dataset. The reason why we chosen pandas is as it supports to load data as DataFrames, since DataFrames are preferred for two dimessional dataset. Our dataset was prepared in .CSV file extension. So, process of loading our dataset from a CSV file into a Pandas DataFrame is achieved using the “read_csv” function in Pandas and

encoding scheme we used is “latin1” as wolaytta language uses Latin script. Also, df.fillna() function is used to fill any missing values in the dataset.

```
import pandas as pd
df = pd.read_csv("wolaytta_NER.csv", encoding="latin1")
df = df.fillna(method='ffill')
df.head()
```

Figure 5. 2 Implementation of loading dataset

5.2.3 Cleaning Dataset

In our dataset, Cleaning involves elimination of irrelevant characters, punctuation marks, emoji’s so on. In the below implementation snippet shows two functions to clean our dataset. The first, clean_text() function strips numbers and emoji’s if exist. The second Remove_punc() function removes any punctuation mark from WNER dataset. Finally, after removing unwanted characters and punctuation marks the change is applied on our dataset by the lambda function.

```
"""Cleaning WNER dataset"""
def clean_text():
    df['Word'] = df['Word'].apply(lambda Word: Word.translate(str.maketrans('', '', "
    df['Word'] = df['Word'].str.strip()
clean_text()
```

Figure 5. 3 Cleaning WNER Dataset

5.2.4 Normalizing WNER Dataset

The data that has been cleaned was given to normalizer function. Normalization is crucial task in Wolaytta language that have different word forms, but with the same meaning. To achieve normalization task we have utilized RegExp library for python. Because RegExp (regular expression) library is preferred for searching of particular matching pattern in the dataset and substituting with required one.

```
# Words to be norimalized in wner dataset
word_dic = { "ai": "ay","ai": "ay","oi": "oy","au": "awu"," diya": "de'iya"," daana": "de'ana",
                    "alamiya":"salo-gufantto","katamaa ": "allaana"}

# Regular expression for finding contractions
word_re=re.compile('(%s)' % '|'.join(word_dic.keys()))

# Function for expanding contractions
def normalize(text,word_dic=word_dic):
    def replace(match):
        return word_dic[match.group(0)]
    return word_re.sub(replace, text)

# Expanding Contractions in the actual text
df['Word'] = df['Word'].apply(lambda x:normalize(x))
```

Figure 5. 4 Implementation of WNER dataset normalization

5.3 Implementation of Feature Extraction

In above steps our dataset have passed all loading and preprocessing steps, since the left one is feeding the dataset to selected models in computer understandable way. To attain this goal we used Word-level features with DictVectorizer encoding technique.

5.3.1 Word-Level Features

We have selected them because they are preferred for checking contextual information of particular token. In the function *word2features ()* below, we transform each token into a feature dictionary describing that token's attributes like; lower case version of the token, suffix containing two and three characters of the token, flags to determine upper case of the token, title case (if first character capitalized), whether that particular token was numeric or not and word's POS tag. Additionally, we have attached attributes related to previous and next tokens to determine BOS (beginning of sentence) and EOS (end of sentence).

Also below *Figure 5.5,* shows implementation of feature extraction for selected algorithms.

```
#Word-Level Feature Extraction for Wolaytta NER
def word2features(sent, i):
    word = sent[i][0]
    postag = sent[i][1]
    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        'word[-3:]': word[-3:],#Returns 3 suffices of word
        'word[-2:]': word[-2:],#Returns 2 suffices of word
        'word.isupper()': word.isupper(),#Checks wether word is all capitalized
        'word.istitle()': word.istitle(),#Checks whether word's first character is capitalized
        'word.isdigit()': word.isdigit(),#checks if all are digits
        'word.isalpha()': word.isalpha(), #all letters
        'word.isalnum()': word.isalnum(), #mixture of letters and digits
        'word.punct()':contain_punct(word),#contains punctuation
        'postag': postag,#POS tag
        'postag[:2]': postag[:2],# POS tag

    }
    if i > 0:
        word1 = sent[i-1][0]
        postag1 = sent[i-1][1]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
            '-1:word.isalpha()': word.isalpha(),
            '-1:word.isalnum()': word.isalnum(),
            '-1:word.punct()':contain_punct(word),
            '-1:postag': postag1,
            '-1:postag[:2]': postag1[:2],

        })
    else:
        features['BOS'] = True
```

Figure 5. 5 Implementation of word-level Feature Extraction

Below Fig 5.6 is the implementation of DictVectorizer. The vector function is initialized and training data (X) fit in to transform then our word level features will be converted to dictionary of records. Also our dependent variable is converted to list of array. Since then, our classifiers can understand features as intended.

```
vectorizer = DictVectorizer(sparse=False)
X = vectorizer.fit_transform(X.to_dict('records'))
y = df.Tag.values
classes = np.unique(y)
classes = classes.tolist()
```

Figure 5. 6 DictVectorizer for word-level features

5.4 Implementation of Models

After all pre-processing and required feature extraction that is feed in to classifiers, we can implement all selected algorithms. Below we have depicted our classifier algorithms implementation.

5.4.1 Conditional Random Field Model

CRF is popular probabilistic model for structured prediction. We initiated to use it because of it is current state-of-art model for NER task. Below to see all possible CRF parameters we checked its docstring and selected L-BFGS training algorithm as it have default Elastic Net (L1 + L2) regularization to penalize our model. We have set maximum _ iteration to 100 and all transition from one label to another was set true Boolean value.

```
crf = sklearn_crfsuite.CRF(
    algorithm='lbfgs',
    c1=0.1,
    c2=0.1,
    max_iterations=100,
    all_possible_transitions=True
)
crf.fit(X, y)
```

Figure 5. 7 CRF model implementation

5.4.2 Support Vector Machine Model

From SVM algorithm, we have implemented LinearSVC, as our features are sequential data representation. We preferred LinearSVC because of it uses one vs the rest scheme. The parameters in *LinearSVC (random_state=0, tol=1e-5), random state=0* manages the pseudo random value generation for shuffling the dataset for the dual coordinate descent, we initialized it as 0 because it has no effect on the results. In addition, we set tolerance stopping criteria. 0

```
from sklearn.svm import LinearSVC
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_predict
lsvm = LinearSVC(random_state=0, tol=1e-5)
lsvm.fit(X, y)
svm_pred = cross_val_predict(estimator=lsvm, X=X, y=y, cv=5)
```

Figure 5. 8 Support Vector Machine Model

5.4.3 Random Forest Model

Random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples (in our case parameterized as n_estimator=50) of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. After initializing classifier we have fit and predict its performance with 5-fold cross validation.

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=50, random_state=1)
rf_pred = cross_val_predict(estimator=rf, X=X, y=y, cv=5)
```

Figure 5. 9 Random Forest Classifier implementation

5.5 Summary

In general, in this chapter we have attempted to elaborate the implementation part of the current study. We have included here from importing different python packages through feature extraction techniques utilized in the study up to implementation of the model were raised.

CHAPTER SIX

6. RESULTS AND DISCUSSION

6.1 Overview

In this chapter, the experimental results presented. It discusses in detail the experimental results of WNER model. We have used 5-fold cross validation to measure our models performance. Henceforth, based on the validation result the confusion matrix and classification reports were generated. On better-performed model, we have experimented on combination of different feature sets to check the effect of those features in WNER model.

6.2 Named Entity Features

In current study, selecting better feature plays vital role for identifying which feature sets perform better in Wolaytta language. To do so, series of experiments were undertook to find out most suitable features for NE tagging task. The main features for the Named Entity Recognition task have been identified based on the different combination of available word and tag context (Alemu, 2013). The features also include suffix for all the words, but as current language lacks prefix it is not applicable for us. We have used below feature combinations by interchanging. Nevertheless, those combinations were applied on outperformed model in accordance with below baseline performance.

F= {W_{i-m},…, W_{i-1}, W_i, W_{i+1},…, W_{i+n}, |suffix≤n|, POS tags, Word-shape(all capital, Title case and whether word is only letter), whether the word IsDigit, whether the word alphanumeric, whether the word contained punctuation inside, BOS, EOS, }.

Where F denotes feature set, W_i is the current word, W_{i-1}…. W_{i-m} denote previous m words of current word. W_{i+1},…, W_{i+n} denote next n word of current word (in current study we used window side of 3 for contextualizing current token). In addition, suffix ≤ *n* implies *n* number of suffixes (in our case 2 and 3 characters are chopped from current word) chopped from current word, contextual part of speech tags, word-shape, whether the word is digit, whether the word alphanumeric, whether the word contained punctuation inside, beginning of a sentence and end of the sentence.

6.3 Dataset Labeling Result

As described in section 3.4 we have total of 16,420 dataset for current study from those 4625 are named entities. The dataset was divided to our four annotators. By following annotation guideline prepared for this study, four annotators have annotated their respective data. Then, for the sake of reliability between those annotator, inter annotator agreement value have been calculated using Cohen's Kappa formulae. Because there are named entities that might belong to more than one class, i.e. certain entity name can be Person name as well as Organization or Location name. So, calculating inter annotator agreement value have become mandated task for mitigating label bias problem of the dataset. For instance, for table 6.1 sample annotation we have calculated agreement value below it:

Table 6. 1 Inter annotator agreement calculation

Annotators	Annotator_1					
Annotatot_2		LOC	PER	ORG	O	Total
	LOC	1500	20	19	23	1562
	PER	35	1490	13	18	1556
	ORG	13	17	1485	24	1539
	O	17	20	16	11710	11763
	Total	1565	1547	1533	11775	**16420**

$\mathbf{P_O}$ = 1500 + 1490 + 1485 + 11710 / 16420 = **0.98**

So expected probability that both would say Location, Person, Organization and Other at random is:

P_{Loc} = 1562/16420 * 1565/16420 = 0.096 * 0.095 = 0.0092

P_{Per} = 1556/16420 * 1547/16420 = 0.095 * 0.094 = 0.0084

P_{Org} = 1539/16420 * 1533/16420 = 0.094 * 0.093 = 0.0087

P_{Other} = 11763/16420 * 11775/16420 = 0.71 * 0.72 = 0.51

Overall random agreement probability is calculated as the probability that they agreed on either Location, Person, Organization or Other, i.e.:

$\mathbf{P_e}$ = P_{Loc} + P_{Per} + P_{Org} + P_{Other} = 0.0092 + 0.0084 + 0.0087 + 0.51 = **0.53**

After applying *Equation 3.1* between Annotator_1 and Annotator_2 we got chance agreement of:

$K = (P_O - P_e) / (1- P_e)$

$= 0.98 - 0.53 / 1 - 0.53 = 0.45/0.47 =$ **0.95 (Agreement between Annotator_1 and 2)**

By following the same scenario, we have calculated inter annotator agreement between four annotators. Therefore, agreement of Annotator_1 and Annotator_2 was calculated first, then between Annotator_2 and Annotator_3 at the second term. Finally, agreement of Annotator_3 and Annotator_4 at the third term. Then, the average probability agreement of all annotators was taken. The first result is 0.95, the second result is 0.89, the third result is 0.92, and the average result of annotators is **0.92**. As can be seen from chapter three Table 3.2, according to Kappa's interpretation standard value, we can infer that there is **very good agreement** between our annotators.

Moreover, the above obtained Kappa result show that our dataset have some ambiguities between annotators that need further improvement to reach in perfect agreement between annotators. This was resulted from single entity name belonged to multiple NE classes.

6.4 Performance of Models

6.4.1 Confusion Matrix for Classifier Models

Two of classifier models were evaluated with 5-fold cross validation to achieve better performing classification model. Confusion matrix we described below are in normalized form. Hence, the confusion matrix is plotted with whole NER corpus, so we are able to make predictions on all dataset. Support Vector Machine classifier performed moderately good as we see in *Figure 6.1* below.

Confusion Matrix for SVM Classifier:

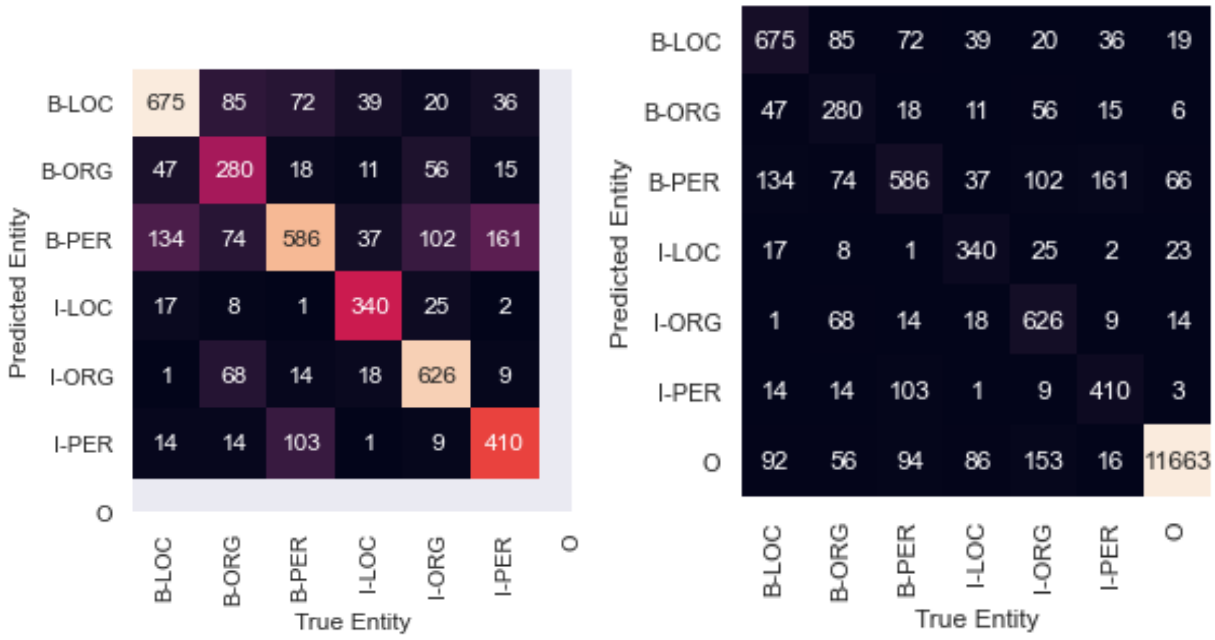


Figure 6. 1 Confusion matrix of SVM without "O" entities (left) and with "O" (right)

As plotted in above confusion matrix we have depicted both Entity classes with respect to "O" (outside of NE) classes. SVM classifier have correctly classified 14,580 words correctly out of 16,420 words. It have miss classified 1840 tokens of averaged fold size. Even from the above confusion matrix we can refer that majority of NEs were classified as "O" entity type because of dominating behavior of non-named entity tokens in our corpus.

Below we have enlisted baseline measured metrics performance of current classifier. Their values were nearly populated.

Table 6. 2 Performance of SVM with all feature sets

Evaluation Metrics	Performance in percent
Precision	88.80%
Recall	88.82%
F1-Score	88.82%

Then, we have given the dataset for Random Forest classifier to see how the data could be populated on the confusion matrix. Unluckily, it have not performed well in our NER corpus, rather performed poorly as we can see in *Figure 6.2* below.

Confusion Matrix for RF Classifier:

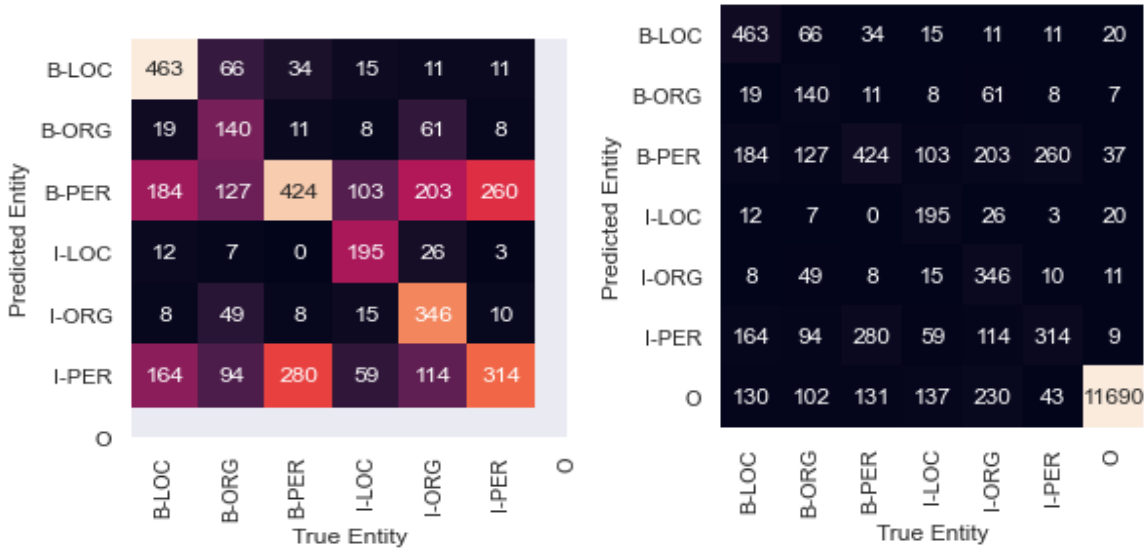


Figure 6. 2 Confusion Matrix of RF without "O" entities (left) & with "O" (right)

As can be seen above, Random Forest is not performing as intended. As, more of NEs were dispersed in the confusion matrix. Even though, we have performed the same fold cross validation as we have did in SVM, but RF model is incompatible with our NER dataset. From the whole dataset of 16,420 words, it have classified correctly 13,572 words and miss classified 2848 words. We can infer from this, as SVM is good at performing on the WNER dataset. Below summarized performance metric evaluation can tell us more, rather than written words.

Table 6. 3 Performance of RF with all feature sets

Evaluation Metrics	Performance in percent
Precision	83.8%
Recall	82.7%
F1-Score	81.9%

Finally, we left with discriminative model but not classifier, rather it is statistical modeling technique used in our structured prediction. The above described models were classifiers models. As, classifiers predict a particular label for a single sample without taking in to account neighboring samples, but CRF can take context into account. CRF[10] is known for implementing dependencies between the predictions it does. So, we have not plotted confusion matrix for CRF, instead plotted state transitions to know how effectively the model transit from one entity to another using eli5 python package. Figure 6.3 depicts state transition learned by CRF model from our dataset.

From \ To	O	B-LOC	I-LOC	B-ORG	I-ORG	B-PER	I-PER
O	3.681	2.067	-1.696	2.494	-1.999	3.154	-2.476
B-LOC	0.375	-0.003	4.851	-0.192	-2.066	-1.619	-1.284
I-LOC	-0.073	-0.206	2.946	0.587	-1.871	-1.595	-2.217
B-ORG	0.089	-0.237	-1.423	-1.181	7.687	-2.917	0.0
I-ORG	-0.362	-1.454	-1.874	-0.798	6.103	-0.598	-3.006
B-PER	-0.121	-0.495	-2.522	-0.43	-1.411	0.503	5.029
I-PER	-0.676	0.122	-2.521	0.0	-1.383	-2.207	4.106

Figure 6. 3 State transition matrix of CRF in WNER dataset.

We can infer from above transition matrix the CRF model well understood succeeding and preceding of entities from our labeled dataset. Though, in the figure above positive valued transitions imply legal and likely tag sequences. From the transition matrix we can infer the model have well fitted with BIO legal tag sequence stated in chapter three Table 3.3. Moreover, performance evaluation result of CRF model shown promising result.

Table 6. 4 Performance result of CRF with all features

Evaluation Metrics	**Performance in percent**
Precision	93.8%
Recall	94.9%
F1-Score	91.4%

[10] https://en.wikipedia.org/wiki/Conditional_random_field

Lastly, in accordance to above evaluation result CRF have yield optimal result than the previous two. Therefore, by using the above CRF model performance as baseline, it would be better to check the effect of different feature combinations.

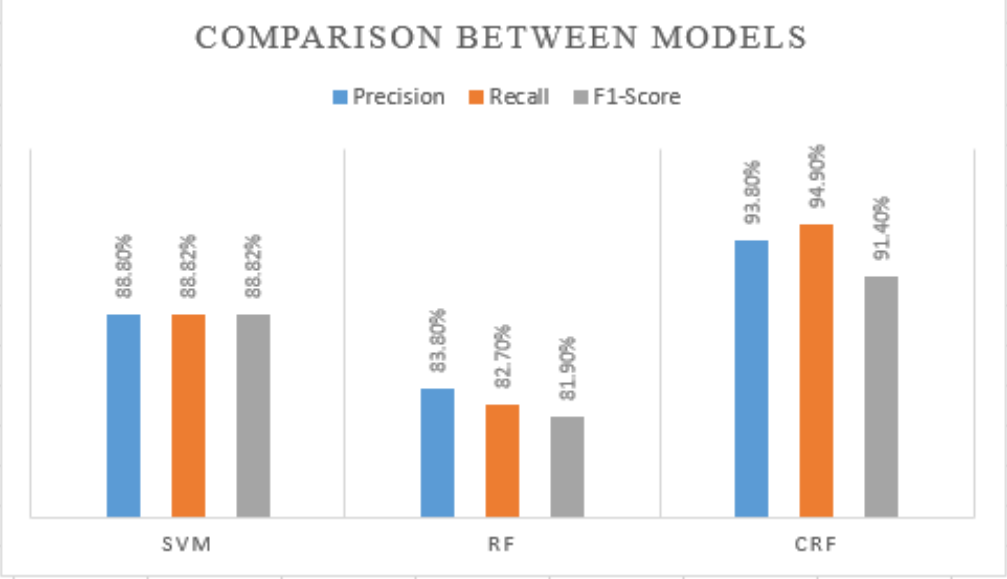


Figure 6. 4 Comparison between models with evaluation metrics

6.4.2 Investigating Feature Influences

At the start, we have evaluated three of selected machine learning model with respect to their fittingness with our dataset. We have chosen CRF model because it well performed in the current study. After, identifying better-performed model that is CRF, we have conducted experimentation with different feature combinations. Here, one feature was dropped at time and evaluation will take place with left features. We have to note that is, if the dropped feature have no any decreasing effect of performance as compared to the baseline result, then the feature is not much relevant for the model. If the omitted feature has degrading effect on baseline experiment, then that feature is crucial for our NER model. As a result in the below section we have combined different feature sets and evaluated their performance. At time when, particular combination of feature performs better than the baseline result luckily, it could be chosen as best feature for our NER model.

6.4.2.1 Impact of POS Feature

Here we have started by investigating the effect of part of speech tag in current study. We have used POS tag as internal feature, as shown in sample-annotated dataset Appendix A. Below we have evaluated our model without POS tag feature.

Table 6. 5 Performance Result without POS Tag feature

Evaluation Metrics	**Performance of:**	
	Baseline(All Feature sets)	**Absence of POS Tag Feature**
Precision	93.8%	91.6 %
Recall	94.9%	93.7%
F1-Score	91.4%	90.6%

We can see clear variations in above comparison of baseline result and absence of POS tag feature. Absence of POS feature decreased precision, recall and F1-Score by 2.2%, 1.2% and 0.8% respectively. We can infer from the above observation as absence of POS tag feature have negative effect on our model's performance.

6.4.2.2 Impact of Word-shape Feature

Here we are going to evaluate the model by removing word-shape feature, but with existence of all others. Below is evaluated model's performance:

Table 6. 6 Performance Result without Word-shape feature

Evaluation Metrics	**Performance of:**	
	Baseline(All Feature sets)	**Absence of Word-shape Feature**
Precision	93.8%	92.2%
Recall	94.9%	92.3 %
F1-Score	91.4%	91.2%

Furtherly, we have investigated the impact of word-shape feature in our model, and shown us slight difference from the baseline result, but not much as POS feature. When seen out of word-shape feature the CRF model have decreased by precision, recall and F1-Score with amount of 1.6%, 2.6% and 0.2% respectively. From this we can conclude as absence of word-shape feature have also negative influence in our study.

6.4.2.3 Impact of Suffix Feature

In this scenario, we have checked whether suffix feature had impact on our model. The following table demonstrates it:

Table 6. 7 Performance Result without Suffix Feature

Evaluation Metrics	Performance of:	
	Baseline(All Feature sets)	**Absence of Suffix Feature**
Precision	93.8%	93.4%
Recall	94.9%	94.7%
F1-Score	91.4%	91.3%

Lastly, we utilized with absence of suffix of current word. Before that, we set number of characters to be chopped from current word. Since, Wolaytta language shares high inflectional behavior, we have used two and three characters of current word suffix. Nevertheless, in the above result without those suffixes shown as no more effect on our NER model. It decreased precision, recall and F1-score of 0.4%, 0.2% and 0.1% respectively from baseline result.

6.5 Summary of Results

Below we have depicted summary of performance evaluation with core three models. In addition, the summary incorporated the different combination of features without performed CRF model. Below table 6.7 and Figure 6.5 summarizes our models performance result.

Table 6. 8 Summary of Classification of Models

Classification Models	Precision	Recall	F1-Score
SVM	88.80%	88.82%	88.82%
RF	83.8%	82.7%	81.9%
CRF	**93.8%**	**94.9%**	**91.4%**
CRF with Different Feature Combinations			
CRF excluding POS tag feature	91.6 %	93.7%	90.6%
CRF excluding Word-shape feature	92.2%	92.3 %	91.2%
CRF excluding Suffix Feature	**93.4%**	**94.7%**	**91.3%**

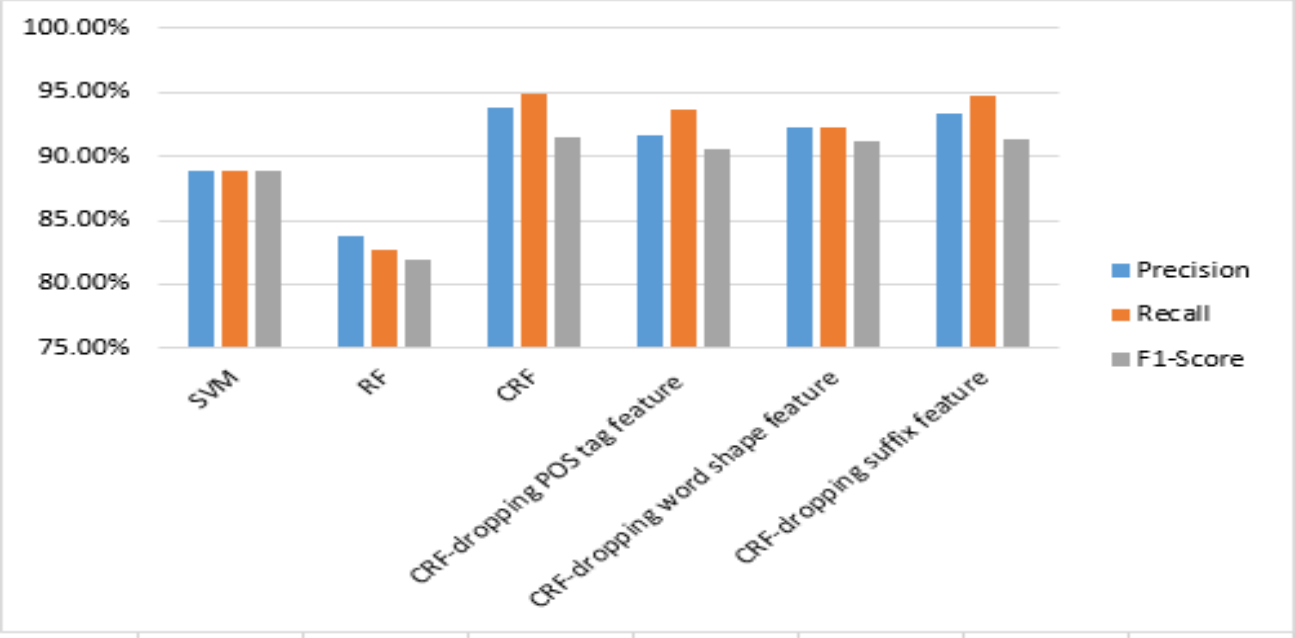


Figure 6. 5 Summary of all results

6.6 Discussion

Developing the effective computational linguistic resource specifically NER model for Wolaytta language was provisioned goal of current study. To our best knowledge, the current work is the first attempt to build Named Entity Recognition model for Wolaytta language by utilizing machine learning. We have undertook various experiments on three machine-learning models CRF, SVM and Random Forest with current NER dataset.

The CRF model that takes context in to account before predicting particular token have better performed in the current study in terms of precision, recall and f1-score. The next out performed model is Support Vector Machine, but RF model is the poorly performed model as compared to the above two. After ensuring better performed CRF model, we have undertaken influence investigation of presence and absence of particular feature value. The feature combination experimentation have helped to determine feature that have positive and negative impact on Wolaytta NER model.

At the time of combined feature investigation, three influencing features were examined. Those examined are POS tag, word-shape and suffix features. In this case, word-shape included four feature sets (all capital, Title case and whether word is only letter). The experimentation was carried out by dropping one of those (POS tag, word-shape and suffix) features at a time. We have ensured that absence of POS tag feature have brought more negative impact in the CRF model. Its absence have reduced higher value of precision and f1-score from baseline result compared to absence of word-shape and suffix feature as can be seen from *Table 6.10*.

Furtherly, absence of word-shape feature is the second influencing scenario we have faced from the model, but not as much as POS tag feature. The feature's absence have thinned down precision, recall and f1-score from baseline result, but its f1-score value was augmented as compared to POS tag feature. Finally, we have assured that absence of suffix feature have no much effect on our model's performance. In contrary, the research work of (Ahmed, 2010) and (Legesse, 2012) clarified as the suffix feature was most influential feature for their case. This might indicate further exploration as why the absence of suffix feature have lower impact on our model.

As we have described in literature review section, even though there was no prior research conducted on NER for Wolaytta language, in comparison to related works for Afaan Oromo (Legesse, 2012) and (Sani, 2015) our current work performance have outshined. In the current study, we have cross-validated our dataset to increase generalization ability of the model, in contrast the above works were conducted with train test split technique, this resulted their model to perform poorly than the current work. For more clarity see below Table 6.11, the comparison of current study with related works.

To conclude, from our experiments we have noticed as language dependent feature like POS tag feature and language independent feature that is word-shape have great positive impact on building NER model for the language. Therefore, we can infer from their result as the combination of POS tag, word-shape and original word itself with context size three played crucial role in successful WNER model development.

Table 6. 9 Comparison of the current work with Related Works

Author	Title and Approach	Well performed model	Context size	Number of unique features used	Model performance
Alemu et al.	NER for Amharic(using ML approach)	CRF	2	5	Achieved Precision, recall and F1-score 76.8% and 84.9% , 80.7% respectively
Ahmed et al.	NER for Amharic(using ML approach)	CRF	3	5	Achieved Precision, Recall and F-score of 72%, 75% and 73.47% respectively
Legesse et al.	NER for Afaan Oromo(using hybrid approach)	CRF	1	6	Achieved Precision, Recall, F1-score of 75.80% , 77.41% and 76.60% respectively
Sani et al.	NER for Afaan Oromo(using hybrid approach)	J48 and C45	1	4	Achieved Precision, Recall and F1-score of 84.12%, 81.21% and 82.52% respectively
Current study	NER for Wolaytta (using ML approach)	CRF	3	8	Achieved Precision Recall and F1-score 93.8%, 94.9% and 91.4% respectively
Major Findings of the study		From Related works gaps listed as (didn't handled acronymic entities, used small amount of NEs for evaluation, limited context size, small feature set, utilization of hand-crafted rules that are unadaptable with new instances) are all handled in the current study.			

CONCLUSION, CONTRIBUTION OF THE STUDY AND FUTURE WORK

Conclusion

In current study, we have designated Named Entity Recognition model for Wolaytta language. To attain our objective, data had been gathered from different sources from those WWRS, WFBCRS and WLLD in Wolaytta Sodo University are main ones. Our collected data need to be labeled in CoNLL's 2002 NER standard way for the current model. As the novelty of the study newly prepared NER corpus was used in our work. Our corpus prepared consists 16,420 words from those 4525 were NEs. The corpus have nearly populated NE instances for three of our classes Person, Location and Organization with 1537, 1412 and 1576 respectively to ensure data balance issue.

The preprocessed corpus need to represent in machine understandable form. This was attained by word-level feature extraction with the help of DictVectorizer method. Afterwards, extracted feature sets were feed in to different machine learning models. We used three machine-learning models like Support Vector Machine, Random Forest and Conditional Random Field for our study and evaluated the dataset with 5-fold cross validation to increase the generalization capability of our models.

The model comparison result shown us Conditional Random Field was out performed model for Wolaytta NER problem. By using out performed, we have conducted different word-level feature combinations. After all, feature set investigation we have ensured as word-shape and POS tag features have major positive impact on the built NER model. However, some of feature combination did not performed as we intended.

Moreover, the developed NER model can be applicable in various domain specific computational areas. The model might be incorporated in advanced language-processing applications like machine translation, information retrieval, opinion mining for the language. Not only have that, the model had great significance for mass Medias those using the language by extracting NE from unstructured text input.

Contributions of the Study

There are various contributions for the current study, among those the major ones are enlisted below:

- ✓ Newly prepared Wolaytta NER dataset can be incorporated in other NLP application.
- ✓ The study proposed new NER architecture for the language using state-of-art approach.
- ✓ The built model have great role for higher NLP problems in Wolaytta that were in need of NER as their input.
- ✓ Identified the pattern of Wolaytta NEs and the way to preprocess in machine understandable form.

Recommendation

To our best knowledge, in Wolaytta language NLP problems like Information Retrieval, Question Answering, Machine Translation and Text Clustering are open research areas till now. The current developed NER model can take major part in the above problems. Therefore, we recommend researchers interested to carry out in those areas to incorporate currently developed NER model for their better performance.

Future Work

As, Named Entities carry vital information in certain Wolaytta document. In addition, there may be various NE classes throughout the document. The current work attempted to explore only three basic entities from NE classes. We suggest interested future researchers:

- ♣ To include all NE classes in Wolaytta text for fully-fledged NER model.
- ♣ To examine the current work with deep learning algorithms by increasing the corpus size.
- ♣ In the current study, we have used Part of Speech tag as an internal feature, we recommend future researchers to build independent POS tagger model to incorporate with NER.
- ♣ In addition, we recommend them to check the effect of Wolaytta stemmer in NER.
- ♣ We recommend one to investigate as why suffix feature had less effect on NER model.

* * *

Special Acknowledgement: *This research was funded by Adama Science and Technology University with the grant number* ***ASTU/SM-R/227/21***

REFERENCES

Ahmed, M. (2010). NAMED ENTITY RECOGNITION FOR AMHARIC LANGUAGE. *Masters Thesis*.

Alambo, H. F. (July, 2010). SPEAKER DEPENDENT SPEECH RECOGNITION FOR WOLAYTTA LANGUAGE. *AAU, Masters Thesis*.

Alemu, B. (2013). A Named Entity Recognition for Amharic. *Masters Thesis Addis Ababa University*.

B. Aryoyudanta, T. B. (2016). Semi-supervised learning approach for Indonesian Named Entity Recognition (NER) using co-training algorithm,. *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), Lombok, Indonesia IEEE*, pp. 7-12, doi: 10.1109/ISITIA.2016.7828624.

B.Upendraa, D. A. (2016). KNN TFIDF Based Named Entity Recognition. *Prasad V Potluri Siddartha Institute of Technology, Vijayawada, AP, India, Volume 1, Issue 12* .

Bayu Aryoyudanta, T. B. (2016). Semi-Supervised Learning Approach for Indonesian Named Entity Recognition (NER) Using Co-Training Algorithm. *2016 International Seminar on Intelligent Technology and Its Application, IEEE*, Pp. 7-12.

BELAY, M. T. (2014). AMHARIC NAMED ENTITY RECOGNITION USING A HYBRID APPROACH. *ADDIS ABABA UNIVERSITY, SCHOOL OF INFORMATION SCIENCE, Masters Thesis*.

Breiman, L. (2001). RANDOM FORESTS. *Statistics Department,University of California, Berkeley, CA 94720*.

Chinchor, N. (1998). Overview of MUC-7/MET-2. *In Proceedings of the 7th Message Understanding Conference*.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20(1)*, Pp. 37-46, doi:10.1177/001316446002000104.

David Nadeau, S. S. (2006). A survey of named entity recognition and classification. *National Research Council Canada / New York University*.

DAWED, O. (2020). AMHARIC NAMED ENTITY RECOGNITION SYSTEM USING STATISTICAL METHOD. *BAHIR DAR INSTITUTE OF TECHNOLOGY, Masters Thesis* .

Demissie, D. (2017). Amharic Named Entity Recognition Using Neural Word Embedding as a Feature. *Masters Thesis*.

Dozier, P. T. (1997). Name searching and information retrieval. *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, pp. 134-140.

Fikre, B. (2020). Part of Speech Tagging for Wolaytta Language Using Transformation Based Learning Approach. *IJESC*.

Gamback, U. K. (2018). Named Entity Recognition for Amharic Using Stack-Based Deep Learning. *Springer Nature Switzerland AG 2018*, pp. 276–287, DOI: https://doi.org/10.1007/978-3-319-77113-7_22.

Girma Yohannis, H. S. (2018). Development of Longest-Match Based Stemmer for Texts of Wolaytta Language. *International Journal on Data Science and Technology, Vol. 4, No. 3, 2018, pp. 79-83.*

Gitimoni Talukdar, P. P. (2014). Supervised Named Entity Recognition in Assamese language. *IEEE*, 187-191.

Gitimoni Talukdar, P. P. (2014). Supervised Named Entity Recognition in Assamese language. *2014 International Conference on Contemporary Computing and Informatics (IC3I), IEEE*, 187-191.

Gowri Prasad, K. F. (6 - 8 May 2015). Named Entity Recognition for Malayalam Language: A CRF based Approach. *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), IEEE*, pp.16-19.

Herano, B. (2015). PART OF SPEECH TAGGING FOR WOLAITA LANGUAGE. *Addis Ababa University, Masters Thesis*.

Hirpassa, S. (Mar – Apr 2017). Information Extraction System for Amharic Text. *International Journal of Computer Science Trends and Technology (IJCST), Volume 5* (Issue 2).

Ipalakova, M. (2010). INFORMATION EXTRACTION. *UNIVERSITY OF MANCHESTER, SCHOOL OF COMPUTER SCIENCE*.

Jason P.C. Chiu, E. N. (2016). Named Entity Recognition with Bidirectional LSTM-CNNs. *Honda Research Institute Japan Co.,Ltd and University of British Columbia*.

Jing Li, A. S. (2018). A Survey on Deep Learning for Named Entity Recognition. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*.

Jing Li, A. S. (2020). A Survey on Deep Learning for Named Entity Recognition. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*.

Johannes, P. (February 2004). Multilingual Information Extraction,. *Department of Computer Science, University of Helsinki 15th*.

Konys, A. (2018). Towards Knowledge Handling in Ontology-Based Information Extraction Systems. *22nd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, 2208–2218.

L. Weston, V. T. (2019). Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature. *Lawrence Berkeley National Laboratory, Materials Science Division*, doi.org/10.26434/chemrxiv.8226068.v1.

Legesse, M. (2012). Named Entity Recognition for Afaan Oromo. *Master‟s Thesis, School of Graduate studies, Addis Ababa University*.

Louppe, G. (2015). U N D E R S TA N D I N G R A N D O M F O R E S T S. *University of Liège ,Faculty of Applied Sciences, Department of Electrical Engineering & Computer Science, PhD dissertation*.

Martin, D. J. (2009). *Speech and language processing, An Introduction to Natural Language Processing, Computational Linguistic, and Speech Recognition, 2nd Edition.* Pearson Prentice Hall, New Jersey.

Meziane, A. E. (2011). *Extracting Person Names from Arabic Newspapers.* 2011 International Conference on Innovations in Information Technology.

Michael Franklin, M. P. (2019). A Word Representation to Improve Named Entity Recognition in Low-resource Languages. *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), IEEE*, 333-337.

Mor Geva, Y. G. (2019). Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. *Tel Aviv University Allen Institute for AI, Google Scholar.*

N.Kannaiya Raja, N. B. (September 2019). NLP: Rule Based Name Entity Recognition. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(11).

Nadeau, S. S. (2006). A survey of named entity recognition and classification. *National Research Council Canada / New York University, USA*.

Nigatu, E. (2021). FAKE NEWS DETECTION FOR AMHARIC LANGUAGE. *ASTU, Masters Thesis* .

Nita Patil, A. P. (2020). Named Entity Recognition using Conditional Random Fields. *International Conference on Computational Intelligence and Data Science (ICCIDS 2019), Elsiever*, 1181–1188.

P. K. Mallapragada, R. J. (Nov. 2009). SemiBoost: Boosting for Semi-Supervised Learning,. *IEEE Transactions on Pattern Analysis and Machine Intelligence , Volume: 31*(Issue: 11).

Padmaja Sharma, U. S. (2014). Named Entity Recognition In Assamese using CRFs and Rules. *AI Lab,Department of Computer Science University of Colorado at Colorado Springs, IEEE*, 15-18.

Rey, M. d. (1999). Extraction Patterns for Information Extraction Tasks: A Survey. *Information Sciences Institute and Department of Computer Science, University of Southern California*.

Richa Sharma, S. M. (2020). A deep neural network-based model for named entity recognition for Hindi language. *Neural Computing and Applications*.

Sani, A. (2015). AFAAN OROMO NAMED ENTITY RECOGNITION USING HYBRID APPROACH. *Master Thesis, AAU, Ethiopia*.

Sarkar, S. (Available Online: [http://www.facweb.iitkgp.ac.in/~sudeshna/courses/ml08/crf.ppt]). *Machine Learning, Conditional Random Fields basics.* Apache Friends.

Shaalan, K. (2014). A Survey of Arabic Named Entity Recognition and Classification. *Association for Computational Linguistics*.

SIKDAR, B. G. (2017). Named Entity Recognition for Amharic Using Deep Learning. *IST-Africa 2017 Conference Proceedings*.

SOLOMON T., M. A. (2015). AMHARIC NAMED ENTITY RECOGNITION [ANER] USING SUPERVISED APPROACH. *UNIVERSITY OF GONDAR, Masters Thesis*.

Sottile, M. L. (2000). "The Wolaytta Language" Some Reactions and Reflections. *Bulletin of the School of Oriental and African Studies, University of London, 63*, pp. 407-420.

Sun, W. (2017). Chinese Named Entity Recognition Using Modified Conditional Random Field on Postal Address. *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2017), IEEE*.

Tewodros A. Gebreselassie, J. N. (2018). A Finite-State Morphological Analyzer for Wolaytta. *ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 14–23, https://doi.org/10.1007/978-3-319-95153-9_2.

Tewodros A., G. J. (2018). A Finite-State Morphological Analyzer for Wolaytta. *ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018*.

Thoudam Doren, K. N. (2009). Named Entity Recognition for Manipuri Using Support Vector Machine. *23rd Pacific Asia Conference on Language, Information and Computation*, pp 811–818.

WAKASA, M. (May 2008). A Descriptive Study of the Modern Wolaytta Language. *The University of Tokyo*.

Wenliang Chen, Y. Z. (2006). Chinese Named Entity Recognition with Conditional Random Fields. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing,Google Scholar*, 118-121.

Wikipedia. ([Online] Available: https://en.wikipedia.org/wiki/Named Entity Recognition [Accessed 10 May 2020]). *Named Entity Recognition.* Wikipedia.

APPENDICES

Appendix A: Sample Annotated Dataset of Wolaytta NER

wolaytta_NER *

	A	B	C	D
1	SENTENCE #	WORD	POS_Tag	NER_Tag
2	Sentence :1	Wolaytta	NNP	B-LOC
3		Mootan	NN	I-LOC
4		Kinddo	NN	B-LOC
5		Diidaye	NN	I-LOC
6		Allaanaa	NN	I-LOC
7		Zali"iyaanne	VC	B-ORG
8		Issipetetta	VC	I-ORG
9		Xiffate	NN	I-ORG
10		Keettay	NN	I-ORG
11		Goshshanchchay	NN	O
12		mooretibenna	VR	O
13		muume	JJ	O
14		Tukiyaa	NN	O
15		zali"e	NN	O
16		Eenotay	NN	O
17		de"iyoogeetussi	VC	O
18		bayzana	VR	O
19		koshshiyooge	VI	O
20		qonccis	VV	O
21		.	PUN	O
22	Sentence :2	Oofa	NN	B-LOC
23		Allaanaan	NP	I-LOC
24		dumma	JJ	O
25		dumma	JJ	O
26		danuwaa	NN	O
27		poliidda	VC	O
28		assatu	NNS	O
29		bolli	ADV	O
30		qashshuwaa	NN	O
31		pirdday	VR	O
32		immettidooge	NV	O
33		qonccis	VV	O

wolaytta_NER

Appendix B: Dataset Annotation Guideline and Validation

Appendix B.1: Dataset Annotation Guideline

Wolaytta Named Entity Recognition Dataset Preparation Guidelines

Dear esteemed annotators:

We have built these annotation guidelines for the purpose of Wolaytta NER. In our case, Named-entities are often proper nouns that have upper case initials. Their variations are mainly typographic from other non-named entity types. We consider a named entity the real-world object denoting a unique individual with a proper name. The annotation follows the IOB scheme where O is used to annotate all tokens that are not named entities. B-label and I-label respectively indicate that the token is at the beginning of an NE and inside it. Note that, the current study focused only on three main types of named entities in our dataset. Below are main guidelines for our four entity categories PERSON, LOCATION, ORGANIZATION and Other no-named entities:

1.Person

When the entity refers to individual or collective person (more than one individual) including fictitious persons. Even in the case of a collective person annotation, there must be the presence of a proper name.

Coverage of the type Person:

Considered as Person:

- ✓ real persons
- ✓ religious figures (God)

Not considered as Person:

- ✓ expressions which do not contain a proper name

NB: At the time PERSON name have successive strings use B-PER and I-PER tag scheme.

2. Locations

Location (LOC): address, territory with a geopolitical border such as city, country, region, continent, nation, state or province. Examples of locations, all instinctively marked as LOC in our dataset should be:

2A. Administrative locations: refer to a territory with a geopolitical border including district, city.

2B. Region: refers to internal divisions within a state and includes all units between country and city levels.

NB: Not include their sub types; also not include names given to natural geographical spaces such as mountains, plateaus, deserts. If LOCATION name have successive strings use B-LOC and I-LOC tag.

3. **Organizations**

Organization (ORG): commercial, educational, entertainment,
government, media, medical-science, political parties, police , non-governmental, religious, churches, sports clubs are all included in ORG entity.

NB: If ORG name have successive strings use B-ORG and I-ORG tag scheme.

4. Non-annotated entities

Except above listed three entities should be labeled as Other ("O"), because they are out of our scope.

4.1 Punctuation marks

All punctuation marks attached to named entities are left as separate tokens. They are not annotated except when they belong named entities such as for addresses, acronyms and abbreviations.

Appendix B.2: Prepared Dataset Validation Letter

Our dataset was annotated via four language experts from Wolaytta Language and Literature Department in Wolaytta Sodo University. Authorized body have authenticated final version of annotated dataset in the below letter.

አዳማ ሳይንስና ቴክኖሎጂ ዩኒቨርስቲ

ኮምፒውተር ሳይንስ እና ኢንጅነሪንግ ትምህርት ክፍል

<u>የመጠይቁ አላማ፡</u>

በአዳማ ሳይንስና ቴክኖሎጂ ዩኒቨርስቲ በኮምፒውተር ሳይንስ እና ኢንጅነሪንግ ትምህርት ክፍል ለሁለተኛ ዲግሪ (ማስተርስ) ምርምር ማሟያ የሚሆን በወላይትኛ ቋንቋ ላይ መሰረት ያደረገ ኮምፒተራይዝድ ሲስተም ለመስራት ግብአት የሚሆኑ መረጃ ከእናንተ መሰብሰብ ይታወቃል። በተሰበሰበው መረጃ መሰረት ለ16421 ወላይትኛ ቃላት የስም ስያሜ (Named-Entity Recognition) እና Annotation Guideline ተዘጋጅቶል።

በቅድሚያ ለሚያደርጉልን ትብብር ከልብ እያመሰገንን፤ የእነዚህን ቃላት ትክክለኛነት እና Annotation Guideline እንዲያረጋግጡልን በትህትና እንጠይቃለን፡፡

<u>የመጠየቁ አዘጋጅ፡</u>

ብሩክ ሲዳሞ በኮምፒውተር ሳይንስ እና ኢንጅነሪንግ ትምህርት ክፍል የሁለተኛ ዲግሪ(ማስተርስ) ተማሪ።

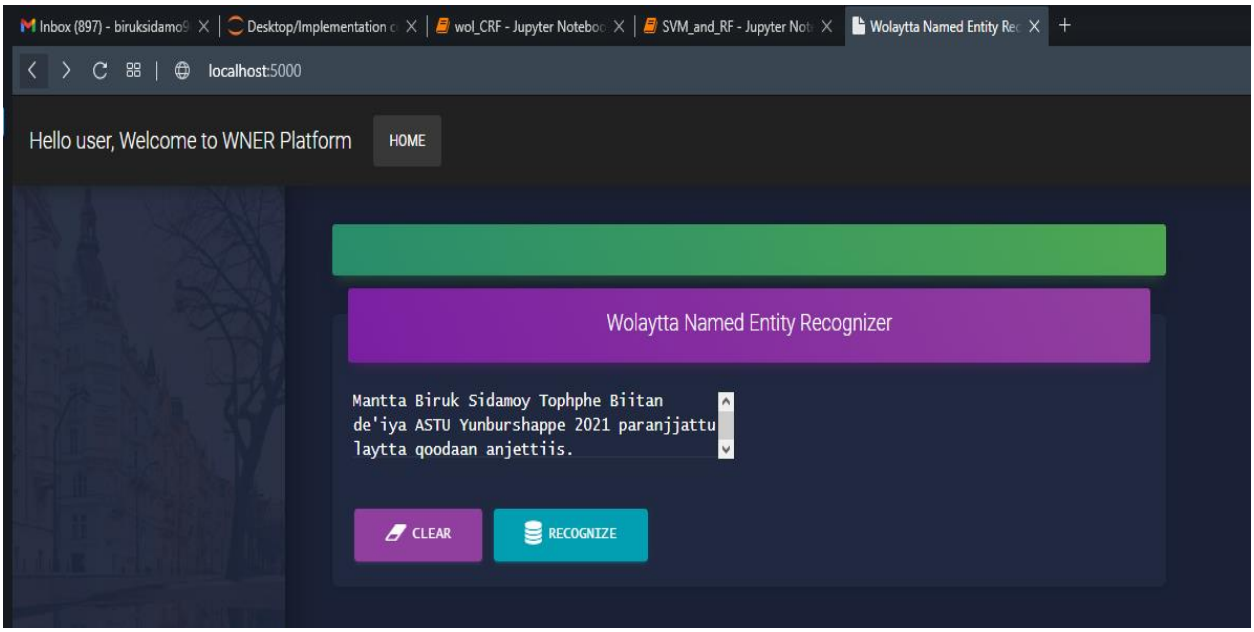
<u>ያረጋጠው፡</u> Warkasha Waaru Alamaaya

1. Alemayehu Waro Warkashe

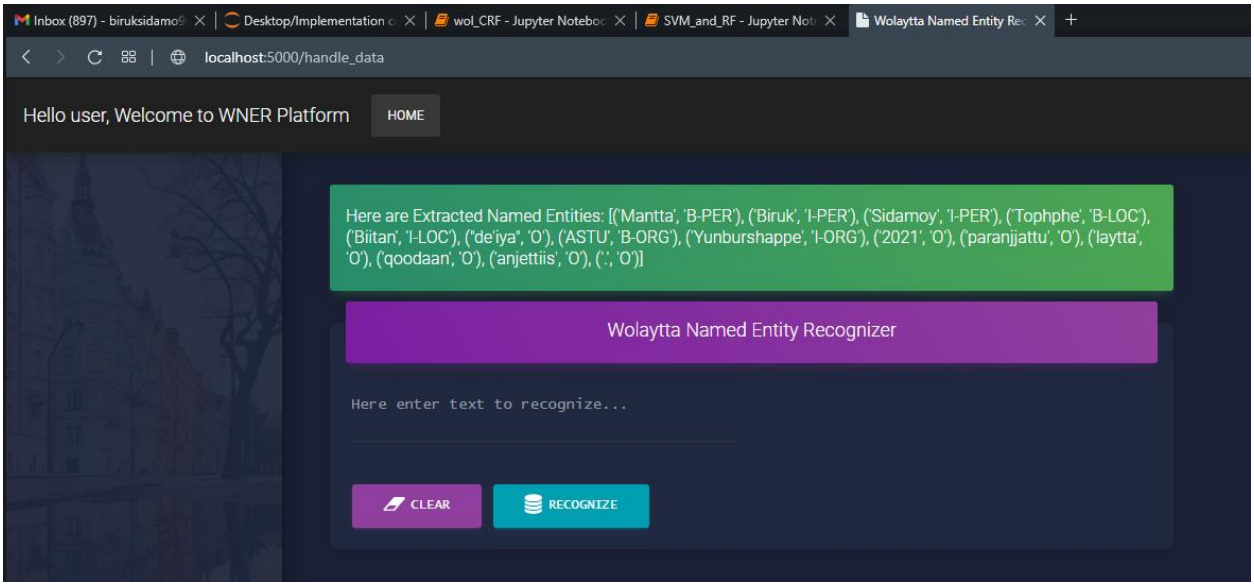
2.

Appendix C: More about Prototyping

Before NE recognition, the user inputs the sentence to be recognized and that sentence would be passed to webserver for actual recognition task. Below *Scenario 1* is the GUI of user input, that have text area in it which is able to accept multi lined sentences. And Scenario 2 is the webserver NE recognized response to user's query.



Scenario 1: Accept User Input Sentence to Recognize NEs



Scenario 2: Named Entity Recognized Webserver Response to User