

Jimma University
Jimma Institute of Technology
Faculty of computing and Informatics



NAMED ENTITY RECOGNITION AND DISAMBIGUATION
SYSTEM

FOR AFAAN OROMO USING SUPERVISED APPROACH

By: Mamo Fideno

A Thesis Submitted to the School of Graduate Studies of Jimma Institute of Technology in

Partial Fulfillment of Masters of Science in Information Technology

Jimma, Ethiopia

February, 2021

Jimma University

Jimma Institute of Technology


Faculty of Computing and Informatics

**NAMED ENTITY RECOGNITION AND DISAMBIGUATION SYSTEM
FOR AFAAN OROMO USING SUPERVISED APPROACH**

By: Mamo Fideno

This is to certify that the thesis prepared by **Mamo Fideno**, entitled **Named Entity Recognition and Disambiguation system for Afaan Oromo using Supervised Approach**, Submitted in partial fulfillment of the requirements for the Degree of Master of Science in *Information Technology* compiles with the regulations of the University and meets the accepted standards with respect to originality and quality.

Approved by board of Examining Committee:

	Name	Signature
Faculty Dean:	_____	_____
Advisor:	<u>Getachew Mamo(PhD)</u>	_____
External Examiner:	<u>Kula Kekeba (PhD)</u>	
Internal Examiner:	<u>Ephrem Taddese</u>	_____
Chair Person:	_____	_____

Jimma, Ethiopia

February, 2021

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with proper citation of sources. I, the undersigned, declare that this thesis has not been presented for a degree in any other university.

Mamo Fidenó

Acknowledgments

First and foremost, praises and thanks to God, for his countless blessings throughout my life and this research.

I would like to express my deep and sincere gratitude to my research advisor, Dr. Getachew Mamo (PhD) and Teferi Kebebew (MSc) for their valuable time, guidance, constructive ideas, and comments which enabled me to gain good research experience.

I have special thanks to Mr. Teshome Daba for your motivation in all of my path. Your words, “You can do more, you have capacity, you will be a great person”, helps me as engine.

I am extremely grateful to my uncles Irko Hunde and Negi Diriba, my aunts Diribe Nagawo, my grand mama Shage Daba and my sister Gadise Bezu for their love, prayers, caring and sacrifices for educating and preparing me for my future. I had great Thanks to my wife Fasika Gebra Mariam and my little brother Abdisa for their all of their effort while I was doing my thesis. They have been my source of strength next to God. Last but not least, I would like to thank Mr. Workineh Tesema, Demeke Endale and all my colleagues.

Abstract

In this research, End to End Named Entity Recognition and Disambiguation problem is addressed by employing a supervised machine learning approach. Feature extraction algorithm had developed; avoiding dependency of Named entity on other natural language processing tasks for classification features. In this paper feature information represented as word vectors are generated from unlabeled Afaan Oromo text. These generated features are used as features for Afaan Oromo Named entity classification and Named entity disambiguation similarity measurements.

A corpus of 10000 sentence had been collected and annotated for Named entity recognition. Word embedding had trained for this paper from 4 million sentences. Knowledge base of 1000 unique entities with their context had been developed for named entity disambiguation.

Conditional Random Field had trained using word embedding as feature for Named entity Recognition. Context based similarity measurement had been implemented for named entity disambiguation. Cosine, Euclidean distance and Jaccard coefficient similarity had tested for context similarity measurement between target and candidate entity context.

From the experiments the highest F-score achieved for Named Entity Recognition was 82.3% using the CRF classifier. The result is similar to state of the art. However, the feature extractor is unsupervised and don't depend on other NLP application. The highest accuracy named entity disambiguation was 62.93% with named entity recognition and 74.21% with data set which had been annotated by human being.

Keyword: Named Entity Disambiguation, Named Entity Linking, Named Entity Recognition and Disambiguation

Contents

Declaration	i
Acknowledgments	iii
Abstract	iv
List of Figure	viii
List of Tables	ix
Abbreviations	x
Chapter One	1
Introduction	1
1.1. Background	1
1.2. Statement of the Problem	4
1.3. Objectives	5
1.3.1. General Objective	5
1.3.2. Specific objective.....	5
1.4. Methodology	5
1.4.1. Research Design.....	5
1.4.2. Literature Review.....	6
1.4.3. Corpus and Data collection	6
1.4.4. Tools and Approach.....	6
1.4.5. Evaluation	7
1.5. Scope and Limitation of the Study	8
1.6. Application of Results	8
1.7. Thesis organization	8
Chapter Two: Literature Review	9
2.1. Theoretical Background	9
2.1.1. Named Entity Recognition.....	9
2.1.2. Named Entity Disambiguation.....	10
2.1.3. Application of Named entity recognition.....	12
2.1.4. Application of Named entity Disambiguation	14
2.1.5. Overview of Afaan Oromo Language.....	16
2.1.6. Approaches of Named entity Recognition	19
2.1.7. Approaches of Named entity Disambiguation	21
2.1.8. Features for Named Entity Recognition.....	23

2.1.9.	Conditional random field (CRF)	23
2.1.10.	Distributed representation of words	25
2.1.10.1.	<i>Skip-gram model</i>	25
2.2.	Related Works	27
2.2.1.	Named entity Recognition	27
2.2.2.	Named Entity Disambiguation	30
2.2.3.	Summary	33
CHAPTER THREE		34
SYSTEM DESIGN		34
3.1.	Introduction	34
3.2.	Architecture	34
3.3.	BIO Annotated Corpus	35
3.4.	Raw Text	36
3.5.	Knowledge Base (KB)	36
3.6.	Preprocessing	37
3.8.	Feature extraction	39
3.9.	Recognition Phase	40
3.9.1.	Model Builder	40
3.9.2.	Prediction	40
3.10.	Named Entity Disambiguation Phases	40
3.10.1.	Candidate selection	41
3.10.2.	Disambiguation	42
Chapter Four: Experiment		44
4.1.	Data Set and preparation	44
4.2.	Development Tools	44
4.3.	Evaluation Metrics	44
4.4.	Baseline Experiment	45
4.5.	Result and Discussion	45
4.5.1.	Word2vec	45
4.5.2.	Named entity Recognition	46
4.5.3.	Named Entity Disambiguation	48
Chapter Five: Conclusion and Future Work		51
5.1.	Conclusion	51

5.2. Future work	51
Reference	53
Appendix	56

List of Figure

FIGURE 2. 1 SKIP GRAM MODEL	26
FIGURE 2. 2 CBOW MODEL ARCHITECTURE.....	26
FIGURE 3. 1 AONERD ARCHITECTURE	35
FIGURE 4. 2 RESULT OF CRF USING DIFFERENT N-ESTIMATOR.....	47
FIGURE 4. 3 DIFFERENT SIMILARITY MEASUREMENT ALGORITHM RESULT WITH NER	49
FIGURE 4. 4 DIFFERENT SIMILARITY MEASUREMENT ALGORITHM WITH GOLD DATASET	50

List of Tables

TABLE 4. 1 RESULT OF THE FIRST WORD2VEC FOR WORD SIMILARITY MEASUREMENT	46
TABLE 4. 2 RESULT OF SECOND WORD2VEC FOR WORD SIMILARITY MEASUREMENT	46
TABLE 4. 3 DATA SET USED FOR NER.....	47
TABLE 4. 4 RESULT FROM CRF PEER EACH CLASS	48
TABLE 4. 5 OVERALL RESULT OF AONER	48
TABLE 4. 6 TESTING DATA SET FOR NAMED ENTITY DISAMBIGUATION.....	48

Abbreviations

NER- Named Entity Recognition

AONER – Afaan Oromo Named Entity Recognition

NERD- Named Entity Recognition and Disambiguation

NLP- Natural Language Processing

RDF- Resource Description Framework

KB- Knowledge Base

NEL – Named Entity Linking

NED- Named Entity Disambiguation

AONERD- Afaan Oromo Named Entity Recognition Disambiguation

POS – Parts of Speech

CRF – Conditional Random Field

BOW -Bag of words

VSM- Vector space model

PER- Person

LOC- Location

ORG- Organization

TP- True Positive

TN- True Negative

FP- False Positive

FN- False Negative

Chapter One

Introduction

1.1. Background

Human being uses a language to communicate. It enables human to formulate and communicate ideas. It can be in the form of speech or text. Speech is transmission of information in the form of sound signal; while text is signs and symbols that represent sounds or utterances. Languages that used by human being are known as natural language while artificial or programming languages are used by machines.

Natural Language Processing (NLP) is a subfield of Artificial Intelligence; which enables computer to analyze and synthesize spoken and written human languages. The main goal of NLP is to get computers to perform useful tasks involving human languages, tasks like human to machine communication, improving human to human communication or processing of text or speech [1].

There are two main reasons why machines or computer agents need to understand human languages: (i) to communicate with human beings with speech and (ii) to enable human to acquire knowledge from written language. NLP is the field of designing techniques and algorithms that process human languages. It takes text or a speech language in one form and converts them into other form of the language or other language.

For instance, Machine Translation is one of the NLP applications that improve human to human communication by translating conversation between two different language speakers like Afaan Oromo to English or vice versa. Machine translation takes a text or a sound in one language, let us say English, and convert it into another language, let us say Afaan Oromo. Thus, to translate the text or sound of one language into another, the machine should have to learn syntax and semantics of the language.

Language syntax processing involves analyzing and synthesizing grammar of the language. It could help to understand how natural languages are aligned with grammatical rules. These are sentence splitting, lemmatization, parts of speech tagging, stemming, chunking, morphological segmentation and syntactic parsing.

The other part of a language is semantic, which is the meaning that the text or speech convey. Semantic processing involves analyzing and synthesizing the meaning and interpretation of text or speech. This includes natural language generation, information extraction, word sense disambiguation, question answering and document summarization.

The increase of electronic records on web makes it difficult to search for specific information from huge amount of data on the internet. It was difficult to search structured fact from unstructured text. Information retrieval and extraction had been developed to solve this problem.

Information Extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured document[2]. For instance, populating database and knowledge base from text. It is the process of analyzing unstructured text to extract predefined entities, events and relation. IE is the process of extracting semantic content from unstructured text[2]. Semantic content includes named entities and events in the text.

Semantic web is the field which tries to enable computer to understand unstructured text on web. NLP is the complementary fields for semantic web. One of the NLP applications in semantic web is the extraction of structured data from unstructured text. Semantic web needs structured data to be accessed by machine. These data should have to be represented in knowledge base in the form of RDF triplet. The triplet consists subject, predicate and object [2]. Both subject and object are real world entities while predicate is the relation between them. The extraction of that triplet from unstructured text is done in NLP. The tools are named entity recognizer, named entity extractor, named entity disambiguation, and relation extraction.

Named entity recognition is the process of identifying and classifying proper names given in a text into predefined groups such as location, person, organization, and time expression[3]. The term named entity was used for the first time on the sixth conference of message understanding which was intended to refer to unique identifier of entities.

The first step of information extraction is to detect the entities in the text[2]. The detection process is done using named entity recognition (NER). NER is the task of information

extraction; which consist of identifying and classifying some type of information elements known as named entity.

Named entity disambiguation is the process of associating the recognized entity to unambiguous entries in a knowledge base. It is the task of disambiguating pre-identified named entities towards a certain KB. It is different from NER, in that it has to compare the context of the recognized entity with the context of the candidate in knowledge base[4].

Named entity disambiguation (NED) is a crucial step in web mining, data mining and semantic web search. NED and named entity linking are the same process with a little difference. Named Entity Linking (NEL) is the process of creating links between the two mentions while the NED is the process of identifying whether they are similar or not. NEL internally uses NED, to identify the mentions similar to candidate entity. NEL includes both named entity recognition and disambiguation.

NED is closely related to Word Sense Disambiguation. The former links mention to instance in open knowledge base like DBPedia; while the later links mention to open word context in knowledge base like WordNet. Both of them need to identify synonym and polysemy[5]. Named entity disambiguation concerns with named entities while word sense disambiguation deals with any words, which have more than one meaning depend on context.

The task of NER is different from that of NED. However, both of them are classification problem. NER has two steps such as recognition and classification. The recognition phase classifies the word as named entity or not while classification categorize recognized entities into predefined classes like Person, Location and Organization. NED is binary classification, whether candidate entity and entity in KB are similar or not. The main difference between NER and NED is that NER classifies entity mention into predefined classes while NED classifies entity mention into entities that are registered in a KB[5].

Named entity recognition and disambiguation is end to end NLP task that identify named entity in text and disambiguate and/or link them to candidate mention in knowledge base. It has two phases. The first phase is entity detection (named entity recognition) and the second phase is disambiguation phase; which compare the similarity between detected named entity

and candidate entity from knowledge base. It plays major role in question answering. It detects entities in queries and documents, then disambiguates whether documents contain answer for the query or not. For instance, for the query “Kilabiin kubbaa miilaa Jimmaa Abbaa Jifaar bara kam hundeeffame?” the system will avoid the document that talks about king “Abbaa Jifaar”. Here the phrase “Abbaa Jifaar” represent two entities depending on the surrounding context: name of a club in Jimma and king who ruled Jimma from 1878 to 1932 G.C. Thus, the system will take the context of the two entities and solve the ambiguities between the entities.

1.2. Statement of the Problem

There are two attempts toward Afaan Oromo Named entity recognition. Those are Named Entity Recognition for Afaan Oromo by Mandafro[6] and Afaan Oromo Named entity recognition by Abdi[7]. The first research had used machine learning approach with CRF algorithm. A corpus of size 23,000 words has been used for training. The training data contains person, location, organization and miscellaneous. The miscellaneous category includes date and time, monetary value and percentage.

The second research claims that CRF has defects in understanding complex structure and incorporated grammar rules in preprocessing stages. The rule-based component has parsing, filtering, grammar rules, white list gazetteers, blacklist gazetteers and exact matching components.

Both of the above attempts require the output of complex feature extractor. The problem is they need complex feature extractors. They are based on the output of other NLP tasks like part of speech tagger and morphological analyzer. Being dependent on other NLP creates performance bottleneck in case of low performing POS tagger or morphological analyzer. To extract a feature for single word many operations have to be performed by feature extractor. Complexity of feature extractor has an effect in response time of the AONER system and makes it difficult to use in practical NLP applications. In other case there is no standard POS tagger and morphology analyzer for Afaan Oromo.

As far as our knowledge is concerned, there is no named entity disambiguation for Afaan Oromo language. The main problem in named entity disambiguation is polysemy and synonymy. Some entities have more than one representation based on context while some

of the entities have more than one. This ambiguity needs to be resolved using entity disambiguation. To develop Named Entity Disambiguation, Named Entity Recognition should have to be developed first since NED takes output from NER. Thus, the problem of named entity recognition should have to be solved before applying its output as input for named entity disambiguation. Thus, in this research the following research had been answered:

RQ1: How to develop automate feature extractor for entity recognition?

RQ2: How to develop NED for Afaan Oromo?

RQ3: what is the impact NER performance on NED performance?

1.3. Objectives

1.3.1. General Objective

The general objective of this research is to investigate named entity recognition and disambiguation for Afaan Oromo system.

1.3.2. Specific objective

- ✓ To investigate the limitation of existing Afaan Oromo named entity recognition model
- ✓ To investigate ways for automatic feature extractor for Afaan Oromo NER
- ✓ To investigate the dependency between NER and NED
- ✓ To prepare corpus and knowledge base.
- ✓ To develop Named Entity Recognition Model
- ✓ To develop Named Entity Disambiguation Model
- ✓ To evaluate the models using appropriate statistical techniques
- ✓ To report the final outputs

1.4. Methodology

1.4.1. Research Design

This research used Experimental research method as the design is tested in different mechanism like changing the features.

1.4.2. Literature Review

Different literatures have been reviewed to conduct this research from journal articles, books, conference proceedings and search engine to identify the gaps, appropriate approaches for NER and NED, and necessity of NER and NED.

1.4.3. Corpus and Data collection

There is no standard corpus prepared for named entity recognition and disambiguation of Ethiopian languages including Afaan Oromo. In addition to this, Afaan Oromo has no knowledge base for named entity disambiguation like DBpedia and Yago. Thus, corpus for both named entity recognition and named entity disambiguation had been prepared by the researchers. The corpus had been collected from online news website such as Fana Broadcasting Corporation (FBC), Oromia Broadcasting Network (OBN), and BBC Afaan Oromo.

Labeled corpus for named entity disambiguation and word2vec had prepared for named entity recognition. In addition to this knowledge base had developed for named entity disambiguation.

1.4.4. Tools and Approach

The common approach used in state-of-the-art is Conditional Random Field (CRF) for named entity recognition. Conditional random field (CRF) is used to train named entity recognition model as it is state of the art for Afaan Oromo named entity recognition. Cosine similarity measurement had been used in for entity disambiguation.

Different python tools and libraries had been used in this research. Numpy and pandas were used for data preprocessing. Gensim is used to implement word2vec. Itertools is used for combining or flattening generated sentence level features together. Sklearn was used for named entity recognition and performance evaluation.

1.4.5. Evaluation

Commonly used metrics for NER such **recall**, **precision**, and **F1 measure** has been used to evaluate NER systems [2]. Under this research those methods are used to measure performance of the developed NER prototype.

Recall is the ratio of the number of correctly labeled responses to the total that should have been labeled[2]. It is the ratio of true positive to total response in the corpus or queries.

$$R = \frac{tp}{tp+fn} \dots\dots\dots \text{equ. 1.1.}$$

Precision is the ratio of the number of correctly labeled responses to the total labeled[2]. It is the ratio of true positive to total response in the corpus or queries.

$$P = \frac{tp}{tp+fp} \dots\dots\dots \text{Equ. 1.2.}$$

Where tp is true positive which means the correct answer given by system, fp is false positive which the wrong answer given as correct by the system, fn false negative are correct answer which are rejected by the system.

F measure is the harmonic mean of the two[2].

$$F_1 = \frac{2PR}{P+R} \dots\dots\dots \text{Equ 1.3.}$$

For named entities, the *entities are* considered as response rather than the word is the unit of response.

Named entity disambiguation is measured using accuracy measurement. This is calculated based on the number of entity that the system linked corrected to knowledge base correctly. This can be measured as follows; n is number of correct linked and N is total number of entities in test set.

$$Accuracy = \frac{n}{N} \dots\dots\dots \text{equ 1.4.}$$

1.5. Scope and Limitation of the Study

This research covers the detection, classification and disambiguation of NAMEX (Person, Location, and Organization). The research design is all about end-to-end named entity recognition and disambiguation. The disambiguation considers only local context of entities in similarity comparison. It doesn't consider the mutual dependency between named entity recognition and disambiguation rather the model accepts the output from NER as input for NED. Only context and disambiguation pages of named entities are considered, Category of named entities doesn't consider under this research.

1.6. Application of Results

Named entity recognition and Disambiguation is the basic tools of NLP applications. It can be used in: knowledge base population, recommender system, chatbot, question and answering, opinion mining, semantic information retrieval, information extraction, web mining, knowledge structuring, entity extraction, relation extraction and machine translation.

1.7. Thesis organization

This work consists five chapters. Chapter two explains theoretical background about NER and NED and related works. Theoretical background explains named entity and named entity recognition, named entity disambiguation, application of named entity recognition and disambiguation, structure of Afaan Oromo, ambiguities in Afaan Oromo, approaches for both named entity recognition and disambiguation and feature for named entity recognition. Related work summarizes works done on named entity recognition for Ethiopian languages and named entity disambiguation for Ethiopian and foreign languages. Chapter three explains the designed architecture for Afaan Oromo Named Recognition and disambiguation. Chapter four is about experiment and evaluation of the model while chapter five presents conclusion, recommendation and future work.

Chapter Two

Literature Review

This chapter provide background for Afaan Oromo named entity recognition and disambiguation. It consists theoretical background; which explains concepts in NERD, and related works that have been done on named entity recognition and named entity disambiguation.

2.1. Theoretical Background

This part clarifies essential hypothesis utilized for development of Afaan Oromo named entity recognition and named entity disambiguation. It explains named entity and named recognition with examples taken from Afaan Oromo Languages. It is followed by the concept of named entity disambiguation and its examples in Afaan Oromo. Application of both named entity recognition and disambiguation are discussed in this part. Following this the current approaches for NER and NED are explained in general. Feature of named entity recognition is explained.

2.1.1. Named Entity Recognition

Named entity recognition is the process of identifying and classifying phrases or proper names given in a text into predefined groups such as location, person, organization, and time expression[3]. For example, the sentences “Pireezidantiin [Yunivarsiitii Jimmaa, Org] Dr. [Jamaal Abbaa Fiixa, PER] jiraattota magaalaa [Jimmaa, LOC] wajjin mari’atan.” contains three named entities: Yunivarsiitii Jimma which is organization name, Jamaal Abbaa Fiixa which is person name and Jimma location name.

The most common or general named entities are proper names which are person, location and organization names. Computational Natural Language Learning had added the fourth entity known as miscellaneous to include other entities like date and money[8]. Named entities can be also task specific like gene which includes protein names or financial assets[2].

The most pertinent information in the document is typically revealed in the names that occur within document[8]. Information extraction systems use NER as their first phase to search documents.

2.1.2. Named Entity Disambiguation

Named Entity disambiguation is the process of associating the recognized entity to unambiguous entries in knowledge bases. Entity linking is also called Named Entity Disambiguation (NED) in the NLP community. Ambiguities in named entities come from either synonym or polysemy[5].

In Synonymy problem, a NEL system needs to match an entity despite its diverse name variations such as abbreviations, spelling variations and nicknames to name a few. Abbreviation is common for Afaan Oromo in organization names. For instance: TOI for “Tajajila Oduu Itiyoophiyaa”, BBO for “Biiroo Barnoota Oromiyaa”, and WALQO for “Waldaa Liqii fi Qusanna Oromiyaa”. Spelling variation is not there in Afaan Oromoo except in case of spelling error while nick name can be found in person name.

The polysemy problem is caused by the fact that multiple entities in knowledge bases (KB)s might have the same name, and this is quite common for named entities. This kind of ambiguities arise from context dependency; which is a big challenge in named entities. For instance, the words “Jimmaa” can be location or one of the clans of Oromo ethnic group depending on the context. Documents contain named entities like person, organization, location, time expression and numbers. These mentions are usually ambiguous due to their polymorphic nature. One name can represent more than one entity depend on the context.

For instance, the word “Abbaa Jifaar” represents sport club, king of Jimma, different bank branches in Jimma like Commercial bank of Ethiopia “Abbaa Jifaar” Branch, Awash Bank “Abbaa Jifaar” Branch and Buna Bank “Abbaa Jifaar Branch”. The word “Gadaa” represents governance system of Oromo people, name of person or name of supermarket. “Itiyoophiyaa” represents name of person or name of a country, Ethiopia. The above, named entities represent different entities from different class person or location in case of “Itiyoophiyaa”, or person or organization in case of “Abbaa Jifaar. In addition to this ambiguity there is ambiguity within the same class. For example, the word Jimma represents

Jimma city and Jimma zone. In the above examples, there is a term “Abbaa Jifaar” which represent different organizations; high school and different bank branches.

The task of addressing the ambiguity problem for named entities is called Named Entity Disambiguation[5]. It maps entity in a given text to the right entities in the given source of knowledge. It is the task of disambiguating pre-identified named entities towards a certain KB[4]. NED aims to automatically resolve mention entities in a document to corresponding entities in a given KB[9]. It is significant for the realization of semantic web and development of NLP applications. Many works were done on the whole text in the document for named entity disambiguation not words around the named entity to be disambiguated [10].

In information retrieval and extraction, machine should have to disambiguate such names based on context in the user query and context in the documents. The responsibility of named entity disambiguation is to rank the similarity of candidate to target entity in such cases.

Named entity disambiguation is essential for semantic text understanding. Automatic text understanding needs to accurately extract potentially ambiguous mention of entities from unstructured text and link them to knowledge bases.

The key challenges in named entity disambiguation are making use of mention context to disambiguate and promoting all the linked entities. It needs design of a good ranking model that computes a reasonable relevance score between candidate entities and corresponding mention based on the information in both the document and knowledge base[9]. Named entity disambiguation gained research attention with the following formal research problem.

Given a document d with a set of mentions $M = \{m_1, m_2, \dots, m_N\}$ and target knowledge base $K = \{e_1, e_2, \dots, e_{|K|}\}$, the task of named entity disambiguation is to find a mapping $M \rightarrow K$ that links mention to correct entity in knowledge base. The output of Named entity disambiguation is set of tuples that are linked to the mention or target entity[11].

2.1.3. Application of Named entity recognition

Named entity recognition can be used in different NLP application like named entity disambiguation, question answering, opinion mining, machine translation, information retrieval, information extraction, text clustering and summarization.

2.1.3.1. *Named entity Disambiguation*

Named entity disambiguation takes the output of named entity recognition as an input. Named entity disambiguation links the recognized named entities to existing knowledge base. Named Entity Recognition helps in candidate selection for disambiguation. For instance, in the sentences “Wajjirri Bulchinsaa Godinaa Jimmaa magaalaa Jimmaa kessatti argama.”, Named Entity Recognition identify that “Wajjirri Bulchinsaa Godinaa Jimmaa” is organization and “Jimmaa” is Location. Candidate selection module accept this search for candidate in knowledge base. For the word “Jimmaa” candidate selection return two entities: “Godina Jimmaa” and “Magaalaa Jimmaa” and avoid “Kilabii Kubba Miila Magaalaa Jimmaa” as it is organization and “Jimmaa” is Location.

2.1.3.2. *Question and Answering*

Question answering is a system that accept user query and give response for the question. It is a kind of IR system in which query is the question and response are short response like sentence, phrase or words. The system gets the users question and search for relevant document in the collection. Then it filters out the documents based on the information in the queries. The similarity between the queries and documents are used to rank the relevant document. Most question answering systems focus on factoid questions, questions that can be answered with simple facts expressed in short texts. The answers to the questions can be expressed by a personal name, temporal expression, or location[2]. Named entity recognition helps in recognizing person, organization and location names in document and queries.

2.1.3.3. *Opinion mining*

The main goal of opinion mining is to develop a system for the extraction of sentiment in documents. It analyzes opinion, appraisal, attitude and emotion given by people toward

products [12]. Opinion mining answers the question who says what about what? From the three questions, named entity recognition answers the questions who and about what. For instance, if somebody gives a comment about one sport club. The person who gives comment and the clubs is named entity. Named entity recognition identifies those named entities in the text. NER uses to identify the consumer of a given opinion or sentiment[2]. It helps to relate the given opinion and the entities.

2.1.3.4. *Machine Translation*

Machine translation take input text from source language and convert into target language text. Named entities don't need translation as they represent entities. For example, the name "Maammoo" in Afaan Oromo is not converted to English or Amharic equivalent. In translation those names should have to be changed into English form "Mamo" or "ማሞ" in Amharic. The names only need to keep the target language grammar; no meaning conversion is need like "sibilaa" in Afaan Oromo is converted into English as metal or "ብረት" in Amharic. In addition to this, named entities are used to disambiguate words in similar surface[13].

2.1.3.5. *Information Retrieval (IR) and Information Extraction (IE)*

Information retrieval (IR) is finding documents of an unstructured nature (usually text) that satisfies an information need from within large collections[14]. The results are documents that are relevant to user's query. Named entity recognition is used in IR to identify named entities both in queries and collection of documents. Following the recognition IR system ranks the documents based on their relevance to user's query using named entities in query and documents.

Information extraction is the process of extracting limited kind of semantic content from text. It turns unstructured data in text to structured data. The structured data can be applicable in many applications like populating relational database. After collecting documents, which is accomplished by IR, the first step of IE is named entity recognition[2]. The other part of IE is relation extraction and event extraction. Both of them needs named entity recognition. Therefore, NER is the main part for IE.

2.1.3.6. Text Clustering

Clustering is used to group data into clusters in which data grouped in one cluster have high similarity[15]. Named entity recognition helps text clustering in ranking document similarities. Identifying named entities in text helps for clustering the text based on entity similarity. For instance, if most of entities in the documents are related to politics, the document will politics document. Thus, all politics entities are grouped together.

2.1.3.7. Summarization

Text summarization aims at compressing long documents into a shorter form that conveys the most important parts of the original document[16]. It is the process of creating summary of certain document that contains the most important message of documents. Named entities help the process as they define the domain of the text. It helps to identify main points in the document.

2.1.4. Application of Named entity Disambiguation

Named entity disambiguation can assist with upgrading the lucidness and add semantics to plain content. It fills in as a pivotal part of numerous natural language applications like information retrieval and extraction, question answering, context analysis and knowledge base population.

2.1.4.1. Information extraction

Information extraction is the process of finding structured data from unstructured text. The first step of information is information extraction is named entity recognition. After recognition recognized entity needs to be linked to existing structured entities. Named entity disambiguation is used in Information extraction to resolve ambiguities between entities in extracted data and target data like relational data base.

2.1.4.2. Query Expansion

NED is a component of query understanding over Knowledge Graphs for annotating entities in queries for further query classification or query interpretation[5]. It helps to expand query by providing different form of named entities like nick name or short form of the names. For example, users can search about Jimma University using JU or Jimma University. As both

names represent single entity, this can be solved using named entity disambiguation. This helps to resolve polysemy in entity expansion.

2.1.4.3. Question answering

Named entity disambiguation can be used question answering in two ways: query expansion and solving ambiguities between names in queries and documents. In query expansion, NED helps in expanding named entities which have more than one names. For instance, if the users ask “Hogganaan TOI enyuu?” (who is director of Ethiopia News Agency), the named entity disambiguation helps to find expanded form of “TOI” (Ethiopia News Agency) which is “Tajaajila Oduu Itiyooophiyaa”. Thus, alternative query will be “Hogganaan Tajaajila Oduu Itiyooophiyaa enyuu?”.

2.1.4.4. Context analysis

Context analysis is processing the text to situate context of text whether it explain about politics, culture, social, economy, philosophy, religion or other issues. It identifies the occasion of the text, aim of the author and intended audience. It identifies doer, receiver and situation of the event. The analyses of content in terms of topics, ideas and categorization get benefit from named entity disambiguation[17]. Linking entities in news article to knowledge bases make better news content analyses. Content analyses can be applied also for social medias.

2.1.4.5. Knowledge base population

Even the largest knowledge bases are far from complete, since new knowledge is emerging rapidly[18]. Knowledge base population is the process of filling incomplete knowledge bases. Information needs to be extracted from unstructured text of different documents, social media and different websites. The process includes relation extraction and entity linking. Relation extraction fills the missed relation between different entities in knowledge bases. Inserting newly extracted knowledge derived from the information extraction to corresponding needs system to map the extracted entity to knowledge bases. Entity linking or named entity liking connects entities in unstructured text to entities in knowledge bases. Named entity disambiguation uses to solve ambiguities between entities in document and knowledge bases.

2.1.4.6. News recommendation

News publishers have decreased disseminating news through conventional newspapers and have migrated to the use of digital means like websites and purpose-built mobile applications. It is observed that news recommendation systems can automatically process lengthy articles and identify similar articles for readers considering predefined criteria[19]. recommendation can be based on entity in the content. For instance, if somebody mostly views news related to Jimma Abba Jifaar sport club, the system will recommend the users as soon as their information related to the club. Named entity disambiguation helps to identify whether the entity is parts of news or not.

2.1.5. Overview of Afaan Oromo Language

2.1.5.1. Background

Ethiopia is one of the multilingual countries; which has more than 80 ethnic groups with diversified linguistic backgrounds. The languages of the country comprise the Afro-Asiatic super family (Cushitic, Semitic, Omotic and Nilotic). Afaan Oromo belongs to an East Cushitic language family of the Afro-Asiatic language super family. It is the most widely spoken language in Ethiopia. It has around 34 million speakers, 34% of the total population of the country, native and the most widely spoken language of Ethiopia. According to Tabor Wami [30], it is also the third most widely spoken language in Africa next to Arabic and Hausa languages. More than two-thirds of the speakers of the Cushitic languages are Oromo or speak Afaan Oromo which is also the third largest Afro-Asiatic language in the world. In the Horn of Africa alone, there are over 45 million native Afaan Oromo speakers[7].

Afaan Oromo is the official language of Oromia Regional state. It is used as a working language of the region, educational language for all subjects except for languages (Amharic and English) in Elementary School (1-8) and as subject in high school (9-10) and preparatory school (11-12).

Some of public universities in Ethiopia including Jimma, Addis Ababa, and Haramaya Universities are offering BA degree and MA degree in Afaan Oromo either in Afaan Oromo Language and Literature or Afaan Oromo Folklore and Literature.

Currently, there are a lot of medias that uses Afaan Oromo as main language of broadcast which includes OBN, OMN, ONN, KMN. The medias that have Afaan Oromo service includes Fana Afaan Oromo (Milito Fana), EBC Afaan Oromo, BBC Afaan Oromo and VOA Afaan Oromo.

2.1.5.2. Syntax of Afaan Oromo Language

In all-natural languages, there is a standardized word order in a sentence. For example, English and French languages have word orders of subject-verb-object. Afaan Oromo and English have differences in their syntactic structure. Afaan Oromo uses subject-object-verb (SOV) form which is similar to Amharic and Japanese languages[6]. Subject-verb-object (SOV) is a sentence structure where the subject comes first, and the object and the verb are second and third elements of a sentence respectively. For instance, in the Afaan Oromo sentence **Ebbisaan kitaaba Maxxanse**. (Ebisa had published the book.), **Ebbisaa** (Ebisa) is a subject, **kitaaba** (the book) is an object and **maxxanse** (published) is a verb.

Afaan Oromo adjectives follow a noun or pronoun; their normal position is close to the noun they modify while in English adjectives usually precede the noun. For instance, **nama cimaa** (strong man), the adjective **cimaa** (strong) follows the noun **nama** (man). There are different rules to word order in Afaan Oromo sentence construction. Understanding of this syntactic structure of sentence can help us to know the relationship between words which in turn leads us to categorize them correctly.

2.1.5.3. Named entity in Afaan Oromo

Named entity is a term or phrase that identifies an object from a set of other objects with in similar attributes [7]. The nature and properties in Afaan Oromo resemble that of NE in English[6]. **Location**, **Person** and **organization** are categorized under proper name while **Miscellaneous** are from different categories such as number, percent and other. The explanation considers only Location, person and organization as per the scope of the research.

Named entities in Afaan Oromo share common characteristics with English language; they are capitalized. Named entities are capitalized whether they are at the beginning, middle or last position of the sentences. This property includes only proper names: Location, organization and Person.

Clue words are also used to show named entities. They are common words that precede named entities. There are different words for person, location and origination. Words that precede person name are Obbo, Aadde, Durbee, Dargaggoo, Insipeektaar, Komandarii, Barsiisaa, Barsiistuu, Doktara, Gargaraa Pirofeesara, Abbaa Gadaa, Abbaa Duulaa and Jeneraala. Some of words that are clues for location are magaalaa, godina, ganda, naannoo, aanaa and aradda. Organization names can be understood from the words like baankii, yunivarsitii, dhaabbata, hospitaala, mana barumsaa, mana sireessaa, waldaa, biiroo, waajjira and warshaa.

2.1.5.4. Ambiguities

Named entity ambiguities in Afaan Oromo can be categorized into three groups as observed from the corpus. These are: single name represents more than one entity (polysemy), single entity with more than one name like nick name (synonym), and different ways of writing for single name composed from more than one word (synonym).

There are named entities in Afaan Oromo which represent more than one entity. For example, the word “Jimma” represents Jimma zone, Jimma city and one of Oromo ethnic clans. The word “Abbaa Jifaar” represents sport club, CBE and Awash bank branch, Airport and king of Jimma.

There are also entities which have more than one name. For example, both “Dandii Qillensaa Itiyooophiyaa” and “Dandii Xiyyaraa Itiyooophiyaa” represent Ethiopian Airline organization. The third ambiguities arise from the ways of writing for names composed from more than one word. Most of these kinds of ambiguities occur in organization and person names. Some locations also have such kind of behavior.

Most of Jimma zone names are composed of two words like “Abbaa Jifar”, “Abbaa Jabal”, “Abbaa Fiixaa” and “Abbaa Macca”. Those names can be written in different forms: “A/Jifaar”, “Abbaa Jifaar”, “Abbaa Jifaar” for “Abbaa Jifaar” and “A/Jabal”, “Abbaa-Jabal”, “Abbaa Jabal” for “Abbaa Jabal”. Religious names like “Haaji Amaan”, “Sheek

Kaadir”,”Gabraa Mariyaam”,”Woldaa Mikaa’el” have such kind of behavior. There are also organization names like “Hooteela Iskaay Laaytitti”, “Hooteela Iskaaylaaytitti” and “Hooteela Iskaay-Laaytitti” which represent single entity, “Itiyoo-teeleekom” and “Itiyoo Teeleekom” represents Ethio Telecommunication Corporation. Location names also show such ambiguities. For instance, “Iluu Abbaa Boor” and “Iluubaaboora” represents the same zone.

2.1.6. Approaches of Named entity Recognition

Named entity recognition can be developed using rule based, machine learning or hybrid (rule based and machine learning) approaches.

2.1.6.1. Rule Based Approaches

Rule based methodologies comprises of a lot of designs utilizing linguistic grammars, syntactic and orthographic features. Early NER systems depended on handcrafted linguistic rules, lexicon, orthographic rules and ontologies[20]. These sorts of frameworks can perform better for confined or specific domain. They can distinguish complex elements that may be hard for learning models. This kind of system has best performance for restricted domain and they can reach perfection. For instance, in the sentence “Dr. Jamal Abbaa Fiixa Pirezidantii Yunivarsiitii Jimmaa ta’uun muudaman.” the word that follows “Dr.” is person name and “Yunivarsiitii” is organization. Different kinds of rule should have to be written for every option that give hints for person, location and organization names. Be that as it may, the main burdens of this methodology are its absence of transportability, power, and significant expense of support in slight difference in information. It is difficult for languages with lack of well-developed linguistic resource. These approaches have extremely high precision but low recall.

2.1.6.2. Machine Learning Approaches

Machine learning algorithms learn from data and make prediction. They operate by building a model from example inputs in order to make data driven prediction or decision rather than following handcrafted rule based programs[21]. Machine learning approaches use a collection of data (corpus) to extract patterns or rules from the data[2].

Machine learning approach is categorized into: Supervised, unsupervised and semi supervised machine learning.

2.1.6.2.1. Supervised machine learning

Supervised machine learning uses a kind of supervision in labeled data. The input for supervised machine learning is tagged with the expected output. The machine learns by observing the pattern in data and their relation to output. The aim is to learn the mapping from input to output according to supervision [22]. It is a kind of learning from example. Classification and regression are the main problems that can be solved by supervised machine learning [23]. Named entity recognition is classification problem, which classify the problem into predefined classes PER, ORG, LOC or MISC. The inputs are tagged using BIO (Beginning inside Outside) tags.

2.1.6.2.2. Unsupervised machine learning

There is no supervision in unsupervised machine learning. It is the opposite of supervised machine learning and there is no labeled data. Unsupervised machine algorithms are used in the absence of annotated data. Clustering is the main problem that can be solved by unsupervised machine learning [23]. Researchers [24][8][25] used unsupervised machine learning to learn feature for named entity learning.

2.1.6.2.3. Semi supervised machine learning

Semi supervised approaches combine labeled and unlabeled data. The main method in Semi Supervised Learning is called bootstrapping which includes small measure of control at the beginning of learning process. The model is trained on an initial set of labeled data then prediction is made on separate set of unlabeled data [13].

2.1.6.3. Hybrid Approaches

Hybrid approaches combine rule based with machine learning approaches. This approach uses both linguistic rules and learned pattern from machine learning algorithm. It takes strong point from both methods. Named entity Recognition for Afaan Oromo by (Abdi,2015)[7] used hybrid to solve limitation of machine learning(CRF) applied by Named entity Recognition for Afaan Oromo[6].

2.1.7. Approaches of Named entity Disambiguation

Over times, Various NED methods have been proposed and they can be categorized based on the feature they use. These are entity prominence, context similarity and entity relatedness.

2.1.7.1. Based on Entity Prominence

The Entity Prominence Service returns statistics about the occurrence of Named Entities in news and blog documents within a certain time period (hour, day, week, month, year)[26]. Methods under this category consider only entity mention and its candidate property. They don't consider the context similarity of the mention. String similarity, popularity and commonness is used as feature[5].

String similarities are based on name string comparison between entities and its candidate. The distance can be measured using Edit Distance, dice and Hamming Distance[5]. It disambiguates based on whether, the entity name of the target entity and name of the candidate are equal or not. In this method the string of both entities should have to be similar. It doesn't consider the context of the entities.

Popularity measure can be used in case of lack of context like single entity. It is domain dependent and can be calculated from Wikipedia view page statics and click popularity[5]. Wikipedia view page statics returns the number of people visited in a given period of time.

Commonness denotes the prior probability of entity which is computed from sense distribution over entity annotation corpora such as anchor text of Wikipedia. If a word or n-gram a appears as an annotation in corpora N times and there are m times linking to the entity E , then the commonness of entity E can be computed as $P(E/a) = m/N$ [5]. Computing entity commonness is dependent on entity annotation corpus which is difficult to obtain and the computed entity commonness probability may only have limited entity coverage because of the incompleteness of the annotation corpora.

2.1.7.2. Based on Context similarity

NED methods based on context similarity discriminate ambiguous entities through measuring similarity between the mention context and the candidate entities. Context

similarity metrics depend on different semantic features in representing contexts and entities. The most intuitive semantic features to represent context are different granularity of texts surrounding the mention, from whole input text to several surrounding words[5].

Entities can be represented by the textual descriptions extracted from knowledge graph (KGs), ranging from the entire Wikipedia page, paragraphs of Wikipedia page, entity summaries, entity abstracts, entity categories, entity types, key phrases, entity titles, and anchor texts[5].

Context-entity similarity can be computed with different similarity metrics based on different models for textual features like Bag of words (**BOW**), Vector space model (**VSM**) and **Distributional vector representation**.

2.1.7.3. Based on Entity Relatedness

Entity relatedness is a special case of context similarity, since entities of other mentions in the input text are used as the semantic feature to represent context. According to the assumption that the input text contains coherent entities from one or few related topics, multiple ambiguous entities are discriminated collectively based on entity relatedness. Such collective disambiguation model is a global model that discriminates all entity mentions jointly. In contrast, NED methods based on entity prominence and context similarity use frequently a local model which considers each entity mention in isolation. The key module of this collective disambiguation model is measuring entity relatedness in order to infer the coherence among candidate entities for all mentions. There are a number of semantic features that can be used to compute entity-entity relatedness based on different type of information sources.

Firstly, semantic contents of entities such as textual descriptions and semantic categories are represented in BOW or VSM to compute entity-entity similarity based on: dot or cosine similarity of entity description or category vectors, topical coherence between entities using overlap of weighted key phrases and topic models and semantic similarity of entity category hierarchies.

Secondly, from entity annotated corpora, entity co-occurrence and entity distribution are used to compute entity-entity relatedness based on the application of distributional

hypothesis which assumes that entities occur in similar contexts is semantically related. Finally, apart from semantic content analysis and distributional analysis, graph analysis is also very effective in measuring entity connectivity in order to compute entity-entity relatedness, given that entities are connected to each other in KGs. Graph analysis measures the entity relatedness based on semantic entity networks using degree analysis or relational analysis. Degree analysis counts the edges connecting entities which only represent occurrence, incoming, or outgoing information, while relational analysis considers semantic meaningful relations between entities. This difference results in different kind of entity relatedness methods[5].

2.1.8. Features for Named Entity Recognition

Features are descriptor for given output. They separate one class from other class. Features for NER includes Parts Of Speech(POS) for target and neighbor word, Embedding of target and neighbor words with given window size, Base phrase chunk, presence of target word in gazetteer, whether the word contain specific prefix or suffix, upper case letter, word shape of target and presence of hyphen[2].

Those features can be categorized as word level and document level list look up feature. Word level features include POS tag, prefix and suffix, word shape and capitalization. List lookup operation includes gazetteer, which is collection of named entities, and word embedding.

2.1.9. Conditional random field (CRF)

Conditional random field is undirected graphical models used to calculate the conditional probability of the values on output nodes given assigned to input values. In what follows, X is a random variable over data sequences to be labeled which is the attributes of words we are going to label, and Y is a random variable over corresponding label sequences which are NE labels PER, LOC and ORG.

A CRF is a framework for building probabilistic models to segment and label sequence data of Natural language processing and Biological sequence like, POS tagging, shallow parsing, named entity recognition gene sequencing, Image processing and computer vision. It is probabilistic models for computing the probability $p(Y|X)$ of possible output $Y = (y_1, \dots, y_n)$

given the input $X = (x_1, \dots, x_n)$ which is also called the observation. There are variants of CRFs like linear-chain CRF and Skip-chain CRF[6].

A special form of a CRF, which is structured as a linear chain, models the output variables as a sequence. This special form of a CRF is known as linear chain CRF and can be formulated as:

$$P(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{j=1}^n \Psi_j(\vec{x}, \vec{y}) \quad (1)$$

Where X and Y respectively represent the sequence of labels and observations that are going to be modelled. $\Psi(X,Y)$ are the different factors corresponding to maximal cliques in the independency graph. The $Z(X)$ is a normalization factor which normalizes the probability distribution to $[0, 1]$. it is computed as:

$$Z(\vec{x}) = \sum_{\vec{y}'} \prod_{j=1}^n \Psi_j(\vec{x}, \vec{y}') \quad (2)$$

It is better to compute conditional probability distribution through conditional independence as they complex in nature. Conditional independence is an important concept used to decompose complex probability distributions into a product of factors, each consisting of the subset of corresponding random variables. This concept makes complex computations efficient. The decomposition is represented by factors of the form $\Psi(X,Y)$:

$$\psi_j(\vec{x}, \vec{y}) = \exp\left(\sum_{i=1}^m \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right) \quad (3)$$

where λ_i represents the parameter estimated for each feature from the training data and m is the total number of features extracted for a single word. Equation 3 is an exponential function (base e). The notation \exp is used since the expression in the bracket is complex to be written as an exponent.

$f_i(y_{j-1}, y_j, \vec{x}, j)$ is a feature function. A feature function is computed from two adjacent labels y_{j-1}, y_j , the whole observation sequence \vec{x} and the current position in the input sequence j . Feature function produces a real value.

In our NERD, NER is trained on Afaan Oromo NE corpus. The trained model will predict the possible NE words from the plain text.

2.1.10. Distributed representation of words

Word representations are representations of words often in vector form which are features of each word. The value of each dimension represents the value for a specific feature. The simplest way of representing words is one hot vector representation. Each word (w) in a vocabulary (V) has a unique index. Then the words are represented by a vector of size $|V|$, in which the index of the word is one and the rest is zero.

One hot representation is easy to understand and implement, but it only considers local context and has limitations. One of the problems with such representation is that, it fails to show correlation between words due to its local nature.

The other word representations are word2vec, glove, Doc2vec, catagory2vec and fast text. Word2vec can be implemented either as skip gram or continues bag of words.

2.1.10.1. *Skip-gram model*

This model accepts a word W_i and predicts the words around the given word (W_i), which are context words (W_{i-2} , W_{i-1} , W_{i+1} , W_{i+2}). Context words do not need to be immediate words. Some words can be skipped within a given window size to look forward and backward from target word. Skip gram model use one hidden layered neural network. The input layer consists of one-hot encoded vector of the vocabulary[27]. Skip gram model neural network is depicted in Figure 2.1.

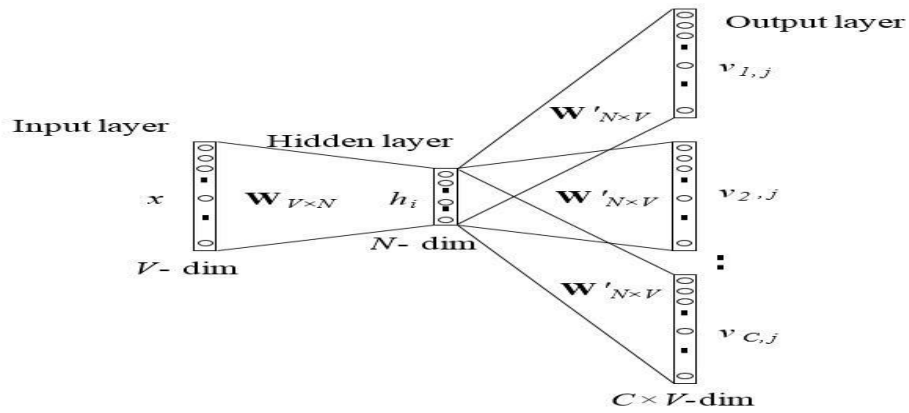


Figure 2. 1 Skip Gram model

2.1.10.2. Continuous bag of words model

Continuous bag of words is reverse of skip gram model. Given the context (W_{i-2} , W_{i-1} , W_{i+1} , W_{i+2}) the task is to predict the word. Continuous bag of words model (CBOW) takes the average of the vectors of the input context words to compute the output of hidden layer, and use the product of the input layer hidden layer weight matrix and the average vector as the output[27]. CBOW model neural network is depicted on Figure 2.2.

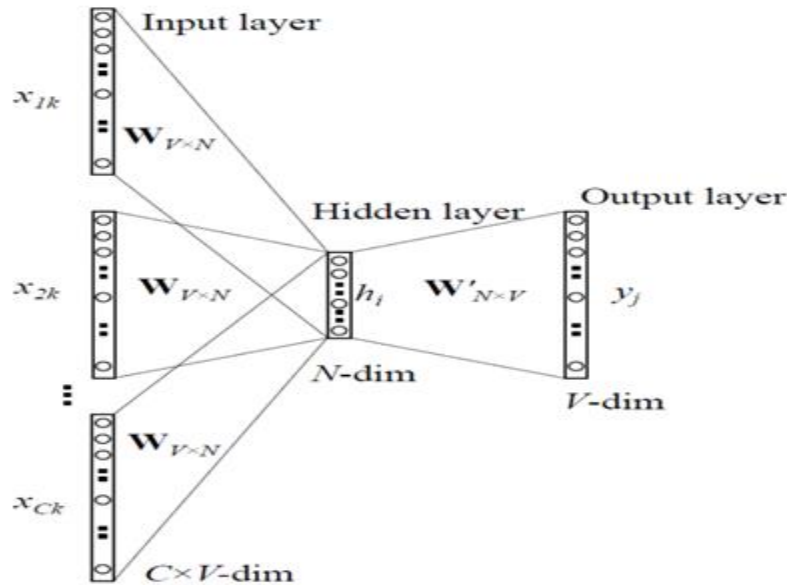


Figure 2. 2 CBOW model Architecture

2.1.10.3. Semantic and syntactic information in word embedding vectors

Word embedding can capture syntax and semantic dependency between words. Syntactic properties of words like in inflectional forms of a word are represented nearest to each other on the continuous vector space. Distributed vector representation of words shows state of the art accuracy on a test set for measuring syntactic and semantic word similarities[27].

Additionally, these vectors can help to get semantic relation between words like city and country, currency to country, opposite, comparative, plural and past tense form of the words can be extracted.

As the model can represent both syntax and semantic of words, using it as feature rich vectors could be used to improve many NLP tasks by substituting manually designed task specific features. Thus, it is used for Afaan Oromo named entity recognition in this research.

2.2. Related Works

This section explains the researches done on Ethiopian and foreign language Named Entity Recognition and foreign language Named Entity Disambiguation. There are two researches done on Afaan Oromo Named entity Recognition. There are no attempts toward named entity disambiguation for Ethiopian Languages. Thus, state-of-the-art named entity disambiguation for English and other foreign languages will be discussed.

2.2.1. Named entity Recognition

2.2.1.1. Named Entity Recognition on Local Languages

Only few researches have been done under Ethiopian languages. Mandefro attempted to develop named entity recognition for Afaan Oromo using CRF [6]. He used a corpus of size 23,000 words has been used for training. The training data contains person, location, organization and miscellaneous. The miscellaneous category includes date and time, monetary value and percentage. The BIO annotation of the corpus was based on CONLL 2002 standard. In this research separate feature extractor had been implemented. It was used to extract word level features: word shape, position, POS tagging, normalized token and morphological feature. Discriminative classifier supervised machine learning had been used

in this research. From this CRF had been chosen for this training. Average performance of 77.41%, 75.80% and 76.60% for precision, recall and F1-score were achieved respectively. However, it depends on other NLP tools such as POS tagger and Morphology analyzer to extract feature.

Abdi sani [7] had used hybrid approach for Afaan Oromo NER. A corpus of the size 27,588 has been used including corpus developed by Mandefro. Rule based approach had been added to the first research to help machine learning in complex features. The rule-based component has parsing, filtering, grammar rules, white list gazetteers, blacklist gazetteers and exact matching components. Machine learning parts had been implemented using decision tree-based algorithm. The best performance achieved from this research was 84.12% precision, 81.21% recall and 82.52% F-Score.

Conditional Random Field machine learning approach had been used for Amharic NER by Ahmed. Word and tag context features, part of speech tag of tokens, prefix and suffix were used as feature. Experiments were conducted on different combination of these features to determine best performing feature sets. For determining these features, four different scenarios were considered. In the first scenario, all features were used. The second scenario used all feature sets except POS tag. In the third and fourth scenarios all features were considered except prefix and suffix respectively. From the experiments, on these scenarios the third scenario achieved highest F-score of 74.61%. Based on the result, the researcher concluded that important combination of features for Amharic named entity recognition is the third scenario which considers all features except prefix feature.

[28] Also used conditional random field machine learning approaches for Amharic named entity recognition. Previous and next word, named entity tag of a word, word pairs, words shape, prefix and suffix features were used for the experiments. The highest F-measure achieved was 80.66%. The researcher recommended using Stanford and ling pipe tools, and using large corpus and adding rule-based component to improve performance of Amharic named entity recognition.

In Hybrid approach is used for Amharic NER in [29], Decision tree (J48) and support vector machine (SVM) are used for classification. Also, a rule-based component having two rules based on presence of trigger words was included in the model. The features used are words,

NE tag of words with window size of two, prefix, suffix, POS tag and nominal feature which indicates whether a word is noun or not. From the experiments the highest F-score achieved was 96.1% for J48 decision tree and 85.9% for SVM.

Lookup operation were used as feature extractor in Amharic NER[13]. The developed system is independent from other NLP application like POS tagger and morphological analyzer. Word vector (Word2Vec) had been used as feature in the research. Different classifier had been tested and deep learning (BLSTM) had outperformed the other. SVM, J48, IBk, random tree, LSTM and BLSTM had been experimented. From the experiments the highest F-score achieved was 92.6% for BLSTM.

2.2.1.2. Named Entity Recognition on Foreign Languages

Adapting word2vec to Named Entity Recognition [24] explored how word vectors built using *word2vec* can be used to improve the performance of a classifier during Named Entity Recognition. Word representation was used to add additional information to classifier. CONLL3 corpus, which is annotated with POS tag, syntactic chunk and named entity tag, was used for both training and testing data. Given a token: POS tag with window 2, syntactic chunk with window 1, uppercase with window 2, conjunction of previous and current token tag and prefix and suffix was used as feature. Word2vec was trained on different size of corpus: CONLL03 and different subset of RCV1 to evaluate the effect of unlabeled corpus. There is also no evidence gained which suggest the direct correlation between corpus size and performance. In general, unlabeled data shows improvement above base line by combining different granularity of clustering.

Named Entity Recognition using Word Embedding as a feature [8] applied word embedding as a feature for named entity recognition (NER) training, and used CRF as a learning algorithm. Glove, Word2Vec, and CCA as the embedding methods had been used in this research. The Reuters Corpus Volume 1 was used to create word embedding and the 2003 shared task corpus (English) of CoNLL was used for training and testing. After comparing the performance of multiple techniques for word embedding to NER, it was found that CCA (85.96%) and Word2Vec (80.72%) exhibited the best performance.

Named Entity Recognition with Word Embedding and Wikipedia Categories for a Low Resource Language [30] used of the proximity of the vector embedding of words to approach the NER problem. The research had been done for one Indian morphologically rich low resourced language known as Bengali. The word vectors obtained from Wikipedia are not sufficient to train a classifier for low resourced language. As a result, they proposed to make use of the distance measure between the vector embedding of words to expand the set of Wikipedia training examples with additional NEs extracted from a monolingual corpus that yield significant improvement in the unsupervised NER performance. The methods had been used based on the hypothesis, word vector belong to the same category like name belong to the same vicinity. The expansion method performs better than the traditional CRF-based (supervised) approach with F-score of 65.4% vs. 64.2%.

Named Entity Recognition Only from Word Embedding's [31] proposes a fully unsupervised NE recognition model which only needs to take informative clues from pre-trained word embedding as the unique feature source. Gaussian Hidden Markov model and Deep Auto encoder Gaussian mixture model had been used to select candidate and their types. Then BiLSTM had been applied. The first layer of the designed model is a two-class clustering layer, which initializes all the words in the sentences with 0 and 1 tags, where 0 and 1 represents non-NE and NE, respectively. The second layer is a Gaussian-HMM used to generate the boundaries of an entity mention with IOB tagging (Inside, Outside and Beginning). The representation of each candidate entity span is further fed into a Deep Auto encoding Gaussian Mixture Model (DAGMM) to identify the entity types. K-means clustering algorithm had been applied to word embedding of the whole vocabulary. BiLSTM had been used as encoder and the output is provided to CRF for NER tagging. Viterbi algorithm had used for decoding process to search the label sequence. The research had been tested on English and Spanish data set. The highest result was gained using LSTM-CRF; which was F-score 68.64%.

2.2.2. Named Entity Disambiguation

Based on the textual feature, Exploiting semantic similarity for named entity disambiguation in knowledge graphs[5] use Information Retrieval (IR) and Latent Semantic Analysis (LSA) to develop the baseline of unsupervised NED approach based on context similarity through

the computation of textual similarity between context and entity descriptions. Semantic Contextual Similarity based NED (SCSNED) relies on contextual word similarity. It improves the baseline that assumes equal importance of contextual words and provides coarse meaning comparison between context and entity descriptions. The SCSNED computes semantic similarity between individual words to offer fine-grained meaning comparison, and uses inverse entity frequency to consider the relative importance of feature words by counting word appearance in descriptions of candidate entities. In order to optimize the performance of SCSNED, both knowledge-based semantic similarity methods relying on semantic knowledge of WordNet and corpus-based semantic similarity methods using word embedding model Word2Vec based on the statistical knowledge from textual corpus had been used. Word similarity had been calculated using cosine similarity. Category2Vec embedding model had proposed to compute word-category similarity for NED in order to provide complement to the word-word similarity feature. Category2Vec learns semantic category and word embedding jointly based on entity abstracts and entity categories, which treats those categories composed by multi-word expressions as a unique semantic unit without separating them into individual words. The highest result gained using context similarity using word2vec was accuracy of 58.5%.

Graph neural entity disambiguation[9] claimed the method in The State-of-the-art for named entity disambiguation CRF do not handle global semantic information. GNERD models global semantic relationship between candidates with the same document. Heterogeneous entity word graph for a document is constructed to encompass the global semantic relationships among the candidate entities in the document. GCN had applied to generate a new set of augmented entity embedding which encode the global semantic relationships among the entities and relevant words by allowing information propagation along the entity-word graph. These embedding of related entities become closer in the embedding space, and thus increase the global coherence of the entities. Then CRF had used to combine the local and global information for collective entity disambiguation. The model had trained using Adam optimization algorithm in an end-to-end Fashion. AIDA-CONLL dataset had been used to experiment the system and the model had improved the average performance of disambiguation was 91.6%.

Research	Authors	Year	Method	F1-Score
Afaan Oromo Named entity Recognition	Mendafro Leggese	2010	CRF	76.6%
Afaan Oromo Named entity Recognition	Abdi Sani	2015	Hybrid (CRF and Rule Based)	82.52%
Amharic Named Entity Recognition	Moges Ahmed	2010	CRF	74.61%
Amharic Named Entity Recognition	Befikadu Alemu	2013	CRF	80.66%
Amharic Named Entity Recognition	Mikias Tadele	2014	Hybrid (Decision Tree and SVM)	DT-96.1% SVM-85.9%
Amharic Named Entity Recognition	Dagimawi Demise	2017	Deep Learning	92.6%
NED				
Exploiting semantic similarity for named entity disambiguation in knowledge graphs	G. Zhu and C. A. Iglesias	2018	unsupervised NED approach based on context similarity	76.5% (Accuracy)

Table 2. 1 Summary

2.2.3. Summary

The researches done for Afaan Oromo Named recognition, depends on other NLP tasks like POS tagger and morphology analyzer. The limitation of POS and morphology analyzer on NER propagates to named entity disambiguation.

Different approaches had been used in Named Entity Disambiguation done of foreign language. Most of them are suitable for developed languages. In case of Afaan Oromo there is only 202 articles on Wikipedia. The other language has high number of articles: (i) English Language has 2,567,509 articles, (ii) 77,444 articles to name few¹. This helps to get entity relation, entity prominence and entity context from Wikipedia statics. The other languages have well developed knowledge graph like Wordnet, which helps to implement graph-based disambiguation.

In case of Afaan Oromo there is no well-developed knowledge graph and some of articles on Wikipedia are empty. Implementing graph-based method needs additional resource. The development of resource needs more time and further study of the language. For instance, wordnet needs more time and linguistic expert to develop. Thus, using entity context similarity based is taken as a solution for this research. From context-based approaches distributional vector representation is used.

¹ https://en.wikipedia.org/wiki/Wikipedia:Multilingual_statistics

CHAPTER THREE

SYSTEM DESIGN

3.1. Introduction

As explained in the statement of the problem there is no attempts on Afaan Oromo Named Entity Disambiguation. In addition to this existing Afaan Oromo named entity recognition depend on other NLP tasks like POS tagger. Thus, after a review of researches done on Afaan Oromo entity recognition one problem identified was design and selection of best features. Our proposed approach aims at automating this process of feature extraction by designing a system that automatically learns features from given unlabeled data. After learning word representations from large unlabeled data, generated word feature vectors are used for training. In addition to this named entity disambiguation is designed; which takes input from named entity recognition.

In this section, we describe an architecture proposed for Afaan Oromo Named Entity recognition and disambiguation (AONERD) that uses automatic features for named entity recognition and context similarity measure for disambiguation.

3.1.1. Architecture

The proposed architecture consists 5 main phases such as preprocessing, word embedding, feature extractor, Recognition and Disambiguation phase. Preprocessing accepts raw text and knowledge base. Word embedding takes preprocessed raw text and train word word2vec. Feature extractor takes word embedding and BIO annotated text to provide feature for named entity recognition. Recognition phase detect names in text and classify as Location, organization and person. Disambiguation phase accepts recognized input from Recognition phase and solve the ambiguities with entities in knowledge bases. It consists Candidate selection, similarity Measurements and ranking.

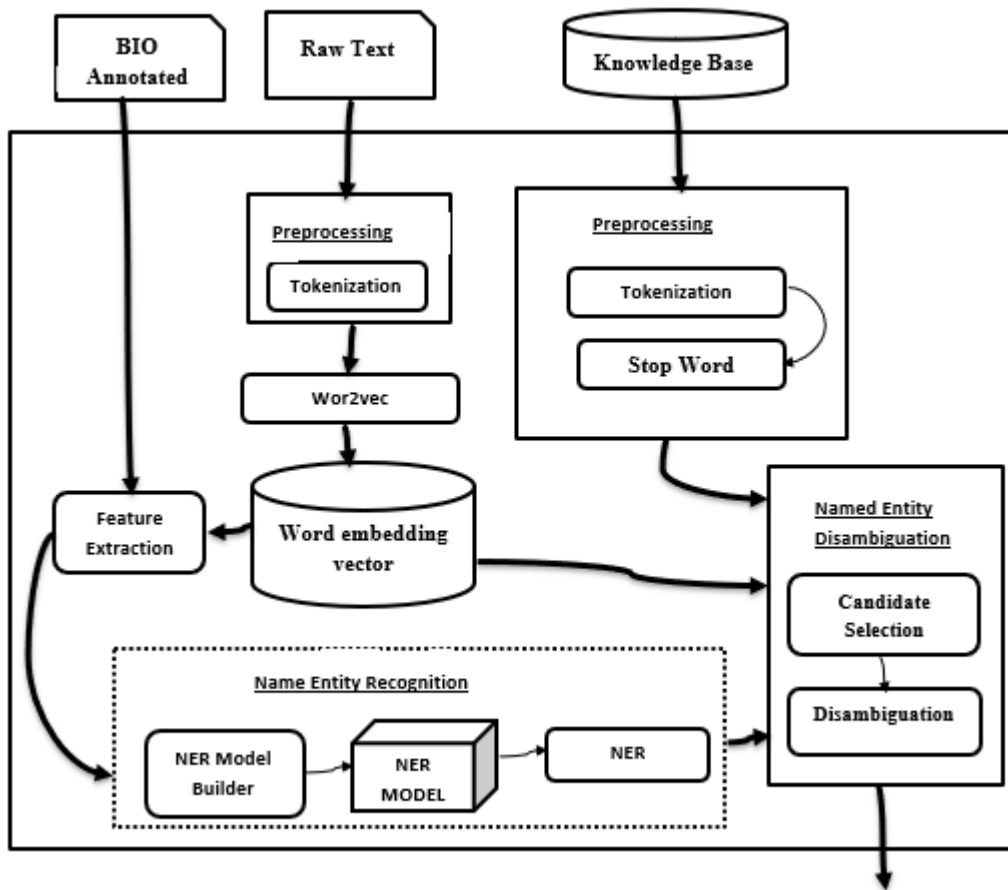


Figure 3. 1 AONERD Architecture

3.2. BIO Annotated Corpus

BIO annotated corpus is NER dataset prepared for named entity training and testing. BIO corpus consists 10000 sentences; which has at least one named entity. The annotation consists B-beginning, I-inside or O-other tags. Beginning tag is used for entity name which has more than one words, to show the begging words. Inside tag is used for both entity with one or more than one words. It is used to show that the word is inside entity name. Other tag is used for non-entity words. Begging and Inside tag are combined with the class of entity like B-PER and I-PER for person class respectively.

For instance, in a single named entity like “Tolosaa” only inside tag is used as (Tolosaa,I-PER), in two word entity like “Tolosaa Bayisaa” both Beginning and inside tag is used as (Tolosaa,B-PER) and (Bayisaa,I-PER).

3.3. Raw Text

Raw text are free text corpus and input sentences which consist entity to be linked. The corpus is used to develop word2vec model. It is collection of sentences from different domain. The total size of the corpus is 4 million sentences collected from different sources such as from Fana Broadcasting Corporate (FBC) Afaan Oromo program (Miltoo Faanaa)², Oromia Broadcasting Network (OBN)³, Oromia Broadcasting Service (OBS) Facebook page⁴, BBC Afaan Oromo⁵ and different social media. There is also additional corpus taken from Habit Project, news scrawled from different medias and social medias.

It helps to develop distributional vector representation of words, which used as feature generation for named entity recognition and similarity measurements in named entity disambiguation.

Input sentences are new sentences which consist entity. The entities in the sentences are recognized by NER and linked to entities in KB using NED. The whole sentence is used as context for recognized entities.

3.4. Knowledge Base (KB)

Knowledge base is collection of entity with their context, disambiguation page; which consists nickname and abbreviation name of entities and id. Context is short summary of entity description extracted from different sources while disambiguation page is list of alternative names for entities. Context of entity is used for similarity measurement and disambiguation page are used for candidate selection.

The entities are taken from free text corpus, which is collected for word2vec. overall, 1000 entities had been collected for knowledge development. The context and disambiguation page had collected from different sources such as Wikipedia, different organization social media pages. The Id of KB had automatically generated after collection of the entities.

² <https://www.fanabc.com/afaanoromoo/>

³ <https://www.bbc.com/afaanoromoo/>

⁴ <https://www.facebook.com/OBNAfaanoromo>

⁵ <https://www.facebook.com/OBSTV>

3.5. Preprocessing

Preprocessing is the first phase of the proposed solution. It is the phases, where necessary process for data preparation is done. There are three process under this module: Tokenization, stop word removal and steaming.

Tokenization is the first process of the module. It splits sentences, phrases, paragraph or entire document into words. It reads the files for training and testing sentences then segment into words. Sentence splitter based on Afaan Oromo grammar had been developed for this research. It split text into sentences, then the sentences into words. The sentence splitter uses end of sentences punctuation marks: {'.', '?', '!'} . The splitter identifies whether period(.) is end of sentence or part words like number (234.56), professional names such as: Dr. Insp. And others. End of sentences punctuation marks are used to split the paragraph or document into set of sentences. The segmented sentences are further split into words. As shown in figure 3.2 below; the splitter had read the file which contain 2 sentences. It splits the content into 2 sentences; using period(.) as end of sentences. However, the period(.) is the word Dr. is not end of sentences; thus, it is taken as a single token. Finally, it tokenizes the sentence into list of words.

```
In [4]: file_to_be_read="C:/Users/Mamo/Desktop/f.txt"
       S=file_into_sentence(file_to_be_read)

Hawwaannisi kun margaa fi baala mukaa mancaasuun nyaata loonirraatti dhibbaa uumeera. Dr. Jamal Abbaa Fiixa Pireezidantii Yuniv
arsiitii Jimmaa ta'uun muudaman.

In [5]: for sent in S:
       print(sent)

['Hawwaannisi', 'kun', 'margaa', 'fi', 'baala', 'mukaa', 'mancaasuun', 'nyaata', 'loonirraatti', 'dhibbaa', 'uumeera']
['Dr.', 'Jamal', 'Abbaa', 'Fiixa', 'Pireezidantii', 'Yunivarsiitii', 'Jimmaa', 'ta'uun', 'muudaman']
```

Figure 3. 2 Tokenization Example

Stop Word removal is the process of removing words that have no effect on the meaning of given sentences. Stop words are taken from compiled appendix of Afaan Oromo stemmer[32]. Stop words are the most common words in the given language. It accepts the tokenized words and remove stop words. List of stop words had used to remove form this process. This module removes the word from the sentences, if the word is in stop word list.

It takes plain text and Knowledge base to remove stop words[32]. It works as shown in figure 3.3 below.

```
In [26]: Sent=remove_stop_words(S[0])

In [27]: print(Sent)

['Hawwaannisi', 'margaa', 'baala', 'mukaa', 'mancaasuun', 'nyaata', 'loonirraatti', 'dhibbaa', 'uumeera']
```

Figure 3. 3 Stop word removal

3.6. Word embedding

The process of automatic feature generation uses preprocessed text in the first step as an input. The input is for this module is large unlabeled data, and this data will be tokenized before it's used for training. This stage is where the words semantic and syntactic relations are learned. The output from this stage is fixed sized vector representation for each word. Word2vec with skip-gram model is used for generating word vectors. After the training process is finished, all the words with their corresponding vector will be logged to an output file. This output file will be the source of features extraction in next stages of our architecture for ANER. It returns the vector of number as shown in figure 3.4 below.

```
In [30]: new_model = Word2Vec.load('model12.bin')

In [35]: print(Sent[1],new_model[Sent[2]])

margaa [-3.123503  0.91492504  1.826967  1.7578983 -1.6030681  0.17806101
-0.96943885 -2.0907357 -0.38939744 -1.2477553  0.5888659  0.78304625
-1.1939757 -0.06298935  0.5847269  2.6542883  0.8078368 -0.56150734
 1.3172876 -0.12311707 -2.9081123 -1.5740666  1.3833668  0.74583566
-1.4233993  0.51638436 -1.2411302  0.05833278  2.2592502  2.854432
-3.5898576  0.39103526 -0.58903986 -1.0223242 -1.5712234  0.64294845
 0.1647197  4.0733814 -4.128477 -0.16527075 -3.0865455 -0.6964632
 0.15566015  0.85322756 -1.4152172 -0.29211825 -1.8668482  1.0066752
-2.6753328 -0.57032084 -0.85290754  1.0972115 -1.5087981 -1.6167551
 1.1401479  0.68446136 -0.6933908 -1.3550067 -0.38991565 -2.2441566
-1.4224021  0.5175658  0.4144019 -1.66735  2.2036736 -3.2946298
 2.5979607  1.1785699 -0.43440115  0.41639692  0.8038153 -3.3507578
-1.4988257 -0.63675684  0.1480924 -0.4887765  1.1848985  3.642131
 2.8690095 -0.16839464 -2.4769692 -5.085747  0.61338854  1.0642388
 2.3252769 -3.7239962 -2.111483 -0.8462377 -0.36473218  2.0999644
 0.03547248 -1.6576962 -0.50717014 -0.55599767  2.69562  2.004338
-2.3883016 -1.3403343 -2.8087049 -0.73153687]
```

Figure 3. 4 Word Embedding result

3.7. Feature extraction

Feature extraction is the process which takes preprocessed words and retrieve it's feature from the above phase. Automatic feature extractor algorithm had developed under this research; which is lookup operation which takes the tokenized words and takes it is vector representation from word2vec model as explained in the algorithm 1 below. For a single, it takes vector representation of ± 2 neighbor words. The retrieved attribute is combined with BIO tag retrieved from named entity tagged corpus. The demo of the algorithm 1 is shown figure 3.5.

Algorithm 1 Feature Extractor algorithm

```
Load saved word2Vec
Read sentence
split sentence into Tokens
for Each token in Tokens do
    for word in  $\pm 2$  do
        | Featureappendword2vec(word)
    end
```

```
In [44]: sent2features(Sent)
        '6.1807394', '-0.57763565', '0.7790672', '1.2267585', '-2.1934385',
        '0.27527472', '5.3123746', '5.072134', '-4.8965535', '-0.16787726',
        '-3.7613106', '5.7947865', '6.0590234', '-2.4775207', '-0.7976476',
        '-4.8292413', '1.5828242', '-0.8748332', '5.2593784', '-1.0315756',
        '0.8263565', '-0.006549924', '-0.27642757', '0.09335227',
        '0.36455446', '-2.7645082', '0.6302891', '-1.6295545',
        '-2.8050418'], dtype='<U32'),
array(['mancaasuun', '-3.123503', '0.91492504', '1.826967', '1.7578983',
        '-1.6030681', '0.17806101', '-0.96943885', '-2.0907357',
        '-0.38939744', '-1.2477553', '0.5888659', '0.78304625',
        '-1.1939757', '-0.062989354', '0.5847269', '2.6542883',
        '0.8078368', '-0.56150734', '1.3172876', '-0.123117074',
        '-2.9081123', '-1.5740666', '1.3833668', '0.74583566',
        '-1.4233993', '0.51638436', '-1.2411302', '0.05833278',
        '2.2592502', '2.854432', '-3.5898576', '0.39103526', '-0.58903986',
        '-1.0223242', '-1.5712234', '0.64294845', '0.1647197', '4.0733814',
        '-4.128477', '-0.16527075', '-3.0865455', '-0.6964632',
        '0.15566015', '0.85322756', '-1.4152172', '-0.29211825',
        '-1.8668482', '1.0066752', '-2.6753328', '-0.57032084',
        '-0.85290754', '1.0972115', '-1.5087981', '-1.6167551',
```

Figure 3. 5 Feature Extraction

3.8. Recognition Phase

Recognition phases detect and classify names in input text. It accepts feature extracted by the above module. Extracted feature are used to build NER model, which is used for prediction. This phase consists model builder and prediction.

3.8.1. Model Builder

Model builder is training process which is the main part of our architecture. It is the estimation of best parameter that gives best prediction. This requires numerical optimization. It uses conditional random field (CRF) machine learning algorithms to build a model. The model estimates λ_i as explained in section 2.9. The input for this process is a training file containing words with their feature vector and named entity tag. After the training data is fed, the model building process starts to form a model considering the features and their named entity tag.

3.8.2. Prediction

The prediction model is the final phase which takes trained model for named entity recognition and feature of target words. The input for prediction phase is the output of feature extractor which is a file containing the word vector of words in a given plain Afaan Oromo text and their vector representation. By taking this input it predicts the named entity tag of a word, therefore the output is target words with their predicted tag.

3.9. Named Entity Disambiguation Phases

Disambiguation phases takes the recognized named entities with their context from the recognition and link them into knowledge base. The first step of this phase is candidate selection, which retrieve candidate entities from knowledge base. After candidate is selected, they pass to disambiguation module which measure similarity between recognized entity and candidates. Similarity measurements used to rank the candidate based on the similarity distance they have. The most similar entity will have high value of similarity or low dissimilarity value.

3.9.1. Candidate selection

Candidate selection is the process which generate entities for the disambiguation process. It takes the output from prediction of Recognition phases as input. For entity e , the candidate selection filters out irrelevant entities in the knowledge base and retrieves a candidate entity set which are relevant to e . There are different mechanisms for candidate selection: Named dictionary based, surface form expansion and search engine based. Surface expansion and search engine based are used in this module. It takes the first letter of the entities and match with the abbreviation. It reads either from abbreviation, if the names are in uppercase, disambiguation and name of the knowledge base. The system also finds its name in disambiguation page. For person names, the systems compare the first name with the entity if the entity name consist only single word. For instance, for person name “Getachoo Maammoo” candidate selection return 2 entities as shown in figure 3.6 below. The first one is Dr. Getachew Mamo;who is instructor at Jimma University , Jimma Institute of Technology , Faculty of computing and the second one is Mr. Getachew Maammoo who is mathematics teacher at Abba Jifar secondary school.

Algorithm 2 Candidate Selection

```
Load KB
Read Entity
Read Entity type
Read all entity with type of Entity type from KB as kbs
initialize candidate=[]
for Each e in kbs do
    if Entity[name] is similar e[name] or Entity[name] ∈ e[disambiguation] then
        | append e into candidate
    end
```

```
In [20]: candidate_selection("Getachoo Maammoo","PER")
Out[20]: [['E0519',
          'Getachoo Maammoo',
          nan,
          "Dr. Getachoo Maammoo bara 2010 hanga bara 2013 dura ta'aa Kompiyutingii turan . ",
          'PER'],
         ['E0520',
          'Getachoo Maammoo',
          nan,
          'Obbo Getachoo Maammoo Mana Barurmsaa Abbaa Jifaar magalaa Jimmaa kessa jirutti barnootaa Herregaa barsiisaa .',
          'PER']]
```

Figure 3. 6 Candidate selection

3.9.2. Disambiguation

This module is responsible to identify the entity which is most relevant to the recognized entity among the candidate based on context similarity. This module uses takes entity repository (KB), named entity dictionary which includes name title and disambiguation page and entity descriptor. Name entity dictionary consist basic information of the entities while descriptor consist context of the entities. It consists similarity measurement and ranking.

Similarity measurement represents semantic distance words in numerical score. It is special case of semantic relatedness which represents commonality of two concepts. Similarity measurement use inverse entity frequency to consider relative importance of feature words and computes semantic similarity between individual words. Similarity can be calculated either in terms of Corpus Based like point wise mutual information and normalized Google distance or predictive based like word2vec. Predictive based similarity measurement is used under this research.

Ranking is the process of selecting the most similar entity depends on similarity measurements. It takes the top similar value from all measurement distance between target entity and candidates. This is done in parallel with similarity, which handle the top value.

For instance, in the sentences “Obbo Getachoo Maammoo mummee barnootaa Herregaatii n ebbifame.”, the disambiguation algorithm link Getachoo Maammo entity with “E0520”; which is Mathematics instructor as shown in figure 3.7 below.

Algorithm 3 Disambiguation

```
Read entity Context as c
Read Entity candidate as E
Initialize Id to NULL
score to 0
for Each e in E do
    read e context as con
    measure similaity between con and c as sim
    if sim is greater than score then
        Assign sim to score
        Assign e[id] to id
    end
end
```

```
In [27]: S=["Obbo Getachoo Maammoo mummee barnootaa Herregaatiin ebbifame ."]
```

```
In [28]: test_link(S)
```

```
Out[28]: [['Getachoo Maammoo',
           'Obbo Getachoo Maammoo mummee barnootaa Herregaatiin ebbifame .',
           'E0520',
           'Getachoo Maammoo',
           'PER']]
```

Figure 3. 7 Disambiguation

Chapter Four

Experiment

This chapter explains the procedures, dataset and tools used under the implementation of the hypothesis.

4.1. Data Set and preparation

Data was collected from different news for this research. Free text corpus had developed for word embedding. This corpus consists 4M sentences. It is collection of text from politics, health, weather, sport, technology, business and agriculture to include all domains. The data that had been collected from news contains different non Afaan Oromo language words in the sentences.

The second corpus is BIO tagged one which is tagged with named entity tag. 10k sentences which have at least one named entity had been selected from the above corpus and annotated manually. The corpus was annotated by following the annotation rule of (Mandefro ,2010) [6] which is similar to CONLL3 under supervision of linguistic expert. The data had divided into training and test data; which is 90% and 10% respectively.

The third corpus dataset is knowledge bases which consist 1k entities. The 20% of the KB had been taken for testing.

4.2. Development Tools

Different tools: libraries and IDE's are used in this research. This includes genism, Sklearn, pandas, NumPy, intercools and anaconda Jupiter notebook.

Genism is library that is used to develop word2vec. We had used different packages from sklearn library for training and evaluation. Iteratools is used to combine data set that had been separated as sentence list into token list after feature generation.

4.3. Evaluation Metrics

A common approach for evaluating machine learning models is through comparison of predicted outputs of the model with that of labeled data by humans. Depending on the similarity between the two, a standard measure used in most researches called F-score can be calculated. F-score is combined measure of precision and recall. Precision is number of

items correctly labeled as belonging to positive class, which is True positive, divided by total number of elements labeled as belonging to positive class which is the sum of True positive and False Positive. Recall is defined as number of true positives divided by total number of elements that actually belong to positive class, which is sum of true positive and false negative.

True positive (TP) measures named entity tags which are predicted by the model that match their labels. False Positives (FP) measures NE tags predicted as positive which is other class. The other one is False negative (FN) are member of class which are missed by the model.

Named entity Disambiguation is evaluated using accuracy measurement. It is calculated from number of entities correctly linked and total number of entities in test set. The entity is correctly linked if the system returns the same id with test data id.

4.4. Baseline Experiment

To test our named entity recognition system we had used research done by Abdi Sani[7] since it is state of the art for Afaan Oromo named entity recognition.

Since there is no named entity disambiguation for Afaan Oromo, we had adapted methods and evaluation from other language. Exploiting semantic similarity for named entity disambiguation in knowledge graphs[5] is used as base line for this research.

4.5. Result and Discussion

Different experiment had conducted to test the designed system. The experiment includes testing accuracy of word2vec, word2vec as feature for named entity recognition, testing performance of different similarity measurements for named entity disambiguation and the joint named entity recognition and disambiguation. The experiment is categorized into three groups: word2vec, Named Entity Recognition and Named entity Disambiguation.

4.5.1. Word2vec

The first experiment is to check whether word2vec can handle context of words or not. Under this experiment, named entities are taken from organization, location and person name. Two models of word2vec vector with 100 size had trained. They are different in data size. The initial model was trained from 17k sentences. This model shows limitation in retrieving

similar words which are in similar context. It combines name of Locations with name of persons. It shows the following result:

Table 4. 1 Result of the first word2vec for word similarity measurement

Words	Top 10 nearest words
Maammoo	Mummee, Makoonnan, Mulaatuu, Abrahaam, Saahilawarqi, Ahmad , Sudaan, Siriil, Sawud
ODP	Dabalata, Aadde, Kabaja,bulchaa,Galaana,hoggansaa,hoji, Buna,Saalvaa,isaa
Arsii	Walloo, Bahaa,Wallagga,Ayyaanichi,Harargee , Godina,Iluu,Gabaa,Baalee,Sabaa

To improve this the data size had increased 4M sentence. The increment of the dataset improves the performance of the word2vec. For the same word the second model respond the following result.

Table 4. 2 Result of second word2vec for word similarity measurement

Words	Top 10 nearest words
Maammoo	Damee, Taddasa,Tsagga,Soolaani,Girmaa, Tafarra,Geetu,Abbi,Kebbedee,Ida’ee
ODP	ABO, Badhaadhina, KFO, Qiniijit, ADWUI, ADP, PBO, Wayyane, WBO, ABO-shanee
Arsii	Baalee, Gaammoo,Gujii, Qellem,Shawaa, Bedellee,Geediyoo, Wallaga, Jimmaa,

The increase of size corpus increases the dispersion of the data, which increase the performance of word2vec. The second model gives related words, which are in same named entity class. This shows that the performance of word2vec depends on the size of the corpus. This also includes the domain of the collected data. If the free text collected for the development is domain specific, it works only for that domain.

4.5.2. Named entity Recognition

The second experiment is to test named entity feature generation. Word2vec had been adapted from State of The Art for other language as feature generator in this research. Word

vector of the given token with window size of 2 had taken for this research purpose. It is lookup operation which takes target word and neighbor word's vector representation.

Conditional random field had trained using word vector as feature. For training the BIO tagged 10k sentences, which has at least one named entity, had used. The total number of named entities is summarized as follow.

Table 4. 3 Data set used for NER

Class	Number of Instance
ORG	5,228
LOC	8,670
PER	5,034
Total=18,932	

CRF takes different parameter: n-estimator, criteria and the others like max-depth, in-sample. n-estimator is total number of trees in the forest which is initially set to 10. Criteria is a function to measure quality of the tree. It can take Gini for Gini impurity, which measure likelihood of an incorrect classification or entropy information gain. It is set to information gain. The other parameters are set to the default value.

The algorithm has tested using different n-estimator: 10, 20, 30, ... , 100. The following result had obtained from the experiments.

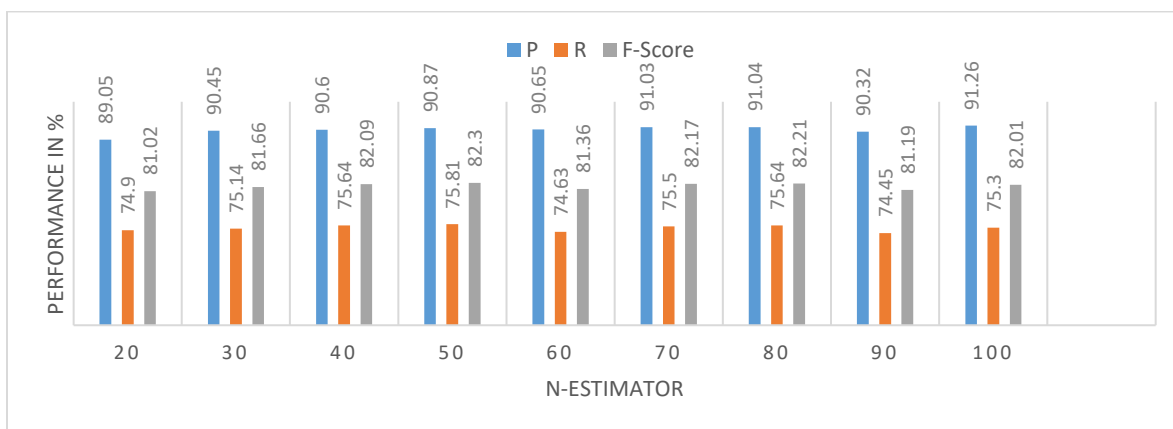


Figure 4. 1 Result of CRF using different n-estimator

As shown on the above as the number of forest (nodes) in CRF increase, F-measure show improvements. It starts to decrease 60 and again increase after 60. It shows best result at 50, which is 82.3.

Table 4. 4 Result from CRF peer each class

Class	P	R	F
Person	93	77	84
ORG	93	76	84
LOC	88	70	78
Weighted average	94.0	94.0	94.0
Accuracy	94.07		

Table 4. 5 Overall Result of AONER

	P	R	F
AONER	90.08	75.81	82.3
Accuracy	94.07		

This shows that the obtained result is almost the same as the base line. Since the word2vec shows the same result as other feature, word2vec can use as automatic feature extractor.

4.5.3. Named Entity Disambiguation

The third experiment is entity context similarity measurement for named entity disambiguation. Three algorithms had tested using Euclidean distance, Jaccard coefficient, Cosine similarity. For this experiment 207 entities had been taken for test from knowledge base of 1000 entities.

Table 4. 6 Testing data Set for Named entity Disambiguation

Class	Number of Instance
ORG	120
LOC	42
PER	45
Total=207	

The test data had prepared with context, their id in knowledge base and entity type. The result is correct if the system responds correct entity from context sentence and with correct id from knowledge base. The context is given into NER model as input. The output from NER is given to NED as input. The candidate for the recognized is selected from KB using candidate selection. The similarity between the recognized entity and candidate is measured using different algorithm depend on their context.

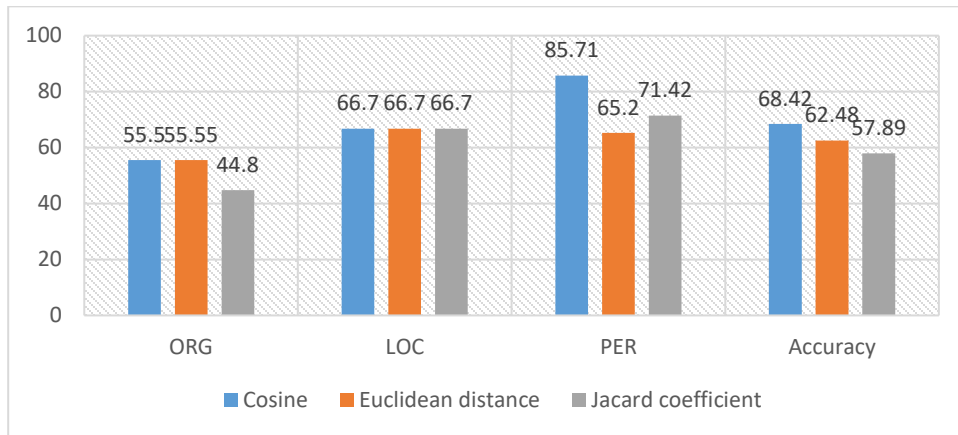


Figure 4. 2 Different Similarity measurement algorithm result with NER

From the three similarity measurements: Euclidean distance, Jaccard coefficient and Cosine similarity, cosine similarity shows best result in this experiment.

The last experiment is done to test the effect of Named entity recognition on named entity disambiguation. Two data set had used to identify the effect of the NER and candidate selection module of NED: is gold data set, which is annotated by human and test data set taken from the output of our model. **Since gold data set have no effect on candidate selection, the change in performance of NED comes from NER on second data set.**

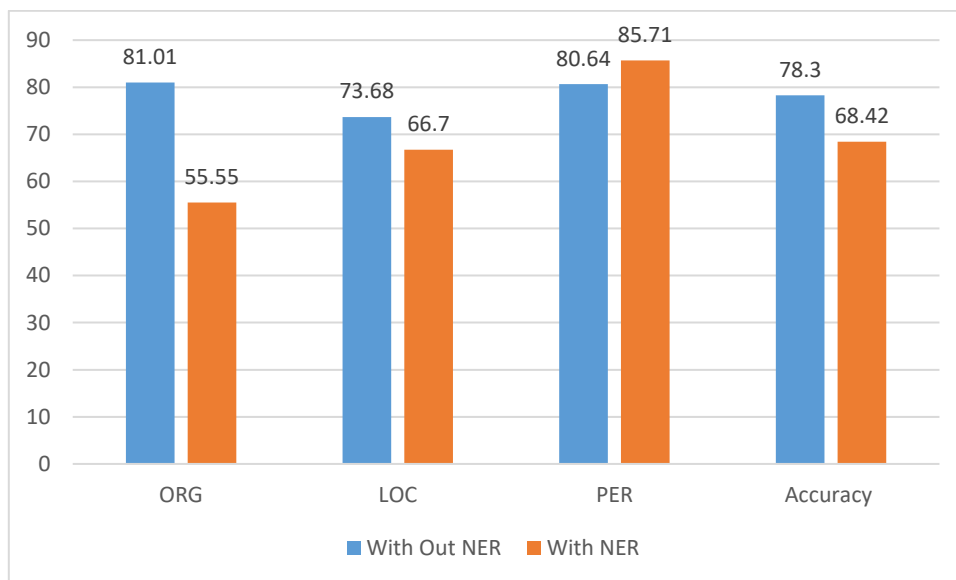


Figure 4. 3 Different Similarity measurement algorithm with human annotated Dataset

The experiments show the performance of named entity recognition has effect on named entity disambiguation.

Chapter Five

Conclusion and Future Work

This chapter explain the conclusion we had we observed in our experiment and future work.

5.1. Conclusion

In this research We trained Word2vec model to generate word vectors that can capture syntactic and semantic relations of words and used it as a feature for our experiments. We have proposed an architecture that uses word embedding as a feature by avoiding usage of manually designed features for Named entity recognition of the model.

Named entity disambiguation has two modules: candidate selection and context similarity measurement. Different approaches had been used for foreign language: Entity prominence based, Context similarity based and Entity relatedness. We had implemented Context similarity-based approaches under this research.

The following point had been concluded from our experiments:

- ✓ Word embedding learned from large unlabeled data can be used as features input for AONER systems. It removes dependency of NER on other NLP applications like POS tagger and Morphology analyzer.
- ✓ Context of the entities had used for similarity measurements. It is implemented using cosine similarity measurements.
- ✓ The experiment shows that the performance of NER have impact on Named entity disambiguation.
- ✓ Improving the performance of named entity recognition, improves the performance of Named entity Disambiguation

5.2. Future work

This research demonstrate end to end named entity recognition. It also shows the performance of Named entity disambiguation depends on named entity disambiguation. Word embedding can be used as feature for named entity recognition using machine learning approaches. But it doesn't consider mutual dependency between named entity recognition

and Named entity disambiguation. The information in named entity disambiguation is used back in named entity disambiguation. Similarity measurement consider only context of the entities. Based on our research we recommend the following points to be investigate in the future:

- ✓ We had applied machine learning approaches. Applying deep learning, we believed that better performance of named entity recognition can be achieved.
- ✓ We had used only text description entity context for similarity measurements. Any researcher can use Category2vec which embed the category of the entity like music, sports, politics, economy to name few.
- ✓ Relation between entity didn't considered under this research. Thus, anybody can use the relationship between named entities in the same context. Wordnet can be used for to extract the relation between those entities.
- ✓ Only Distributional vector representation is used for this research. Any researcher can experiment using BOW and VSM for context similarity. Context similarity consider only

Reference

- [1] D. J. & J. H. Martin., “Speech and Language Processing: An introduction to natural language processing,” *SPEECH Lang. Process. An Introd. to Nat. Lang. Process. Comput. Linguist. Speech Recognit.*, pp. 1–18, 2001, [Online]. Available: <http://www.cs.colorado.edu/~martin/slp.html>.
- [2] D. Jurafsky and J. H. Martin, “Speech and language processing: An introduction to speech recognition,” *Comput. Linguist. Nat. Lang. Process. Edn., Prentice Hall, ISBN*, vol. 10, no. 0131873210, pp. 794–800, 2008.
- [3] Q. Wang and M. Iwaihara, “Deep Neural Architectures for Joint Named Entity Recognition and Disambiguation,” *2019 IEEE Int. Conf. Big Data Smart Comput.*, pp. 1–4, 2019.
- [4] A. Cetoli, M. Akbari, S. Bragaglia, A. D. O. Harney, and M. Sloan, “Named Entity Disambiguation using Deep Learning on Graphs,” 2018.
- [5] G. Zhu and C. A. Iglesias, “Exploiting semantic similarity for named entity disambiguation in knowledge graphs,” *Expert Syst. Appl.*, vol. 101, pp. 8–24, 2018, doi: 10.1016/j.eswa.2018.02.011.
- [6] M. Legese, “School of Graduate Studies Faculty of Computer and Mathematical Sciences Department of Computer Science Named Entity Recognition for Amharic Language Named Entity Recognition for Amharic,” no. October, p. Angeles, L., Advocacy, S., Location, O. (2002)., 2010.
- [7] A. S. Genemo, “SCHOOL OF GRADUATE STUDIES COLLEGE OF NATURAL SCIENCES DEPARTMENT OF COMPUTER SCIENCE AFAAN OROMO NAMED ENTITY RECOGNITION USING SCHOOL OF GRADUATE STUDIES COLLEGE OF NATURAL SCIENCES DEPARTMENT OF COMPUTER SCIENCE,” no. March, 2015.
- [8] M. Seok, H. J. Song, C. Y. Park, J. D. Kim, and Y. seop Kim, “Named entity recognition using word embedding as a feature,” *Int. J. Softw. Eng. its Appl.*, vol. 10, no. 2, pp. 93–104, 2016, doi: 10.14257/ijseia.2016.10.2.08.
- [9] L. Hu, J. Ding, C. Shi, C. Shao, and S. Li, “Knowledge-Based Systems,” *Knowledge-Based Syst.*, no. xxxx, p. 105620, 2020, doi: 10.1016/j.knosys.2020.105620.
- [10] H. T. Nguyen and B. T. District, “Enriching Ontologies for Named Entity Disambiguation,” no. c, pp. 37–42, 2010.
- [11] A. Sun, J. Han, and C. Li, “NeuPL : A ention-based Semantic Matching and Pair-Linking for Entity Disambiguation,” doi: 10.1145/3132847.3132963.
- [12] F. Hemmatian and M. K. Sohrabi, “A survey on classification techniques for opinion mining and sentiment analysis,” *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 1495–1545, 2019, doi: 10.1007/s10462-017-9599-6.

- [13] D. Demissie, T. Submitted, and P. Fulfillment, “Addis Ababa Institute of Technology School of Electrical and Computer Engineering Amharic Named Entity Recognition Using Neural Word Embedding as a Feature Amharic Named Entity Recognition Using Neural,” 2017.
- [14] A. Shiri, *Introduction to Modern Information Retrieval (2nd edition)*, vol. 53, no. 9. 2004.
- [15] A. K. Sangaiah, A. E. Fakhry, M. Abdel-Basset, and I. El-henawy, “Arabic text clustering using improved clustering algorithms with dimensionality reduction,” *Cluster Comput.*, vol. 22, pp. 4535–4549, 2019, doi: 10.1007/s10586-018-2084-4.
- [16] W. Kryściński, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, “Neural text summarization: A critical evaluation,” *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 540–551, 2020, doi: 10.18653/v1/d19-1051.
- [17] W. Shen, “Entity Linking with a Knowledge Base :,” pp. 1–20.
- [18] I. Augenstein, D. Maynard, and F. Ciravegna, “Distantly supervised Web relation extraction for knowledge base population,” in *Semantic Web*, 2016, vol. 7, no. 4, pp. 335–349, doi: 10.3233/SW-150180.
- [19] C. Feng, M. Khan, A. U. R. Rahman, and A. Ahmad, “News Recommendation Systems - Accomplishments , Challenges & Future Directions,” vol. 8, 2020.
- [20] V. Yadav and S. Bethard, “A Survey on Recent Advances in Named Entity Recognition from Deep Learning models,” 2019, [Online]. Available: <http://arxiv.org/abs/1910.11470>.
- [21] M. Moens, *Information Extraction : Algorithms and Prospects*. .
- [22] O. Pentakalos, “Introduction to machine learning,” in *CMG IMPACT 2019*, 2019, doi: 10.4018/978-1-7998-0414-7.ch003.
- [23] Y. Masutani, M. Nemoto, Y. Nomura, and N. Hayashi, *Clinical Machine Learning in Action*. 2012.
- [24] S. K. Sien, “Adapting word2vec to Named Entity Recognition,” no. Nodalida, pp. 239–243, 2015.
- [25] O. Ghiasvand, “Unsupervised Biomedical Named Entity Recognition,” no. August, 2017.
- [26] D. Project, *Studies in Big Data 5 Mastering Collaboration and Decision Making*. 2014.
- [27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.
- [28] B. Alemu, “SCHOOL OF GRADUATE STUDIES SCHOOL OF INFORMATION

- [29] M. Tadele, “August, 2014,” *Addis Abeba Univ.*, 2014.
- [30] A. Das, D. Ganguly, and U. Garain, “Named Entity Recognition with Word Embeddings and Wikipedia Categories for a Low-Resource Language,” vol. 16, no. 3, 2017.
- [31] Y. Luo, H. Zhao, and C. Engineering, “Named Entity Recognition Only from Word Embeddings,” 2018.
- [32] D. T. Gemechu and E. Abebe, “Designing a Rule Based Stemmer for Afaan Oromo Text,” *Int. J. Comput. Linguist.*, vol. 1, no. 2, p. 1, 2010, [Online]. Available: <http://www.cscjournals.org/csc/manuscriptinfo.php?ManuscriptCode=69.70.63.72.41.50.102>.

Appendix

A. LIST of Stop Words

akka , akkam , akkasumas , akum , akkuma , ammo , ammo , ani , booda , booddee , dura , eega , eegana , eegasii , ennaa , erga , fi , garuu , hanga , henna , hoggaa , hogguu , hoo , illee , immoo , ini , innaa , isaa , isaan , iseen , itumallee , ituu , ituullee , jechaan , jechuun , kan , kanaaf , kanaafi , kanaafuu , koo , kun , 'malee , moo , odo , ofii , oggaa , oo , osoo , otoo , otumallee , otuu , otuullee , saniif , silaa , simmoo , sun , tahuullee , tanaafi , tanaafuu , ta'ullee , tawullee , wagga , woo , yammuu , yemmuu , yeroo , yommii , yommuu , yoo , yookaan , yookiin , yookinimoo , yoom

B. BIO Annotated corpus sample

```
In [9]: data.head(100)
```

```
Out[9]:
```

	words	BIO
0	Ministeerri	B-ORG
1	Qonnaa	I-ORG
2	fi	I-ORG
3	Qabeenya	I-ORG
4	Uummamaa	I-ORG
5	hojii	O
6	qabeenya	O
7	uummamaarratti	O
8	magaalaa	O
9	Adaamaatti	I-LOC
10	marii	O
11	gaggeessaa	O
12	jira	O
13	.	O

C. Knowledge Base Corpus Sample

In [13]: `kb.head(20)`

Out[13]:

	id	Entity	Disambiguation	Context	E_TYPE
0	E0001	Tajajjila Oduu Itiyoophiyaa	TOI,T.O.I	TOI'tuu gabaase .	ORG
1	E0002	Laga Xaafoo Laga Dadhii	NaN	Baandii Laga Xaafoo Laga Dadhii	ORG
2	E0003	Laga Xaafoo Laga Dadhii	NaN	Laga Xaafoo Laga Dadhii	ORG
3	E0004	Laga Xaafoo Laga Dadhii	NaN	Magaalaa Laga xaafoo Laga Daadhiitti hiriirri ...	LOC
4	E0005	Jimmaa	NaN	Bulcha itti aanaan godina Jimmaa Obbo Tsaggaay...	LOC
5	E0006	Jimmaa	NaN	Guyyaan HIV/AIDS sadarkaa naannoo Oromiyaatti ...	LOC
6	E0007	Tawaldee Gabramariyam	Tawaldee G/mariyam, Tawaldee Gabra-mariyam	Tawaldee Gabramariyam	PER
7	E0008	Abbaa Jifaar	Kiilnikaa	Kiilnikaa	ORG
8	E0009	Abbaa Jifaar	Mootii	Mootiin Abbaa Jifaar mootii Mootummaa Jimmaa...	PER
9	E0010	Abbaa Jifaar	Mana Barumsaa	Mani Barumsaa Abbaa Jifaar mannen barnootaa ma...	ORG
10	E0011	Abbaa Jifaar	Kilabaa Kuubaa Miilaa	Kilabni Kuubaa Miilaa Adaamaa torbee da'ee Kli...	ORG
11	E0012	Abbaa Jifaar	Bufataa Xiyyara	Bufatnii Xiyyara Abbaa Jifaar ebbifame.	ORG
12	E0013	Adaanach Abeebbee	Adaanach habeebee	NaN	PER