IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 3, Ser. III (May - June 2020), PP 44-61
www.iosrjournals.org

Ethiopian's Prominent LanguageAfaan Oromo Text Chunking Using Conditional Random Fields and Support Vector machines

[1]NegessaWayessaJuka, [2] Dr. Getachew Mamo Wegari, [3]TeferiKebebewTelesa,
[1]*M.Sc. Student, Jimma University, Ethiopia*
[2]*Department of Information Technology, Jimma University, Ethiopia.*
[3]*Department of Information Technology, Jimma University, Ethiopia*

Abstract: *Chunking is a natural language processing (NLP) applicationthatis used toidentify syntactically correlated parts of words in a text. It is the process of separates and segments the text into non-overlapping chunk. This separation and segmentation of text or sentence into chunk is recently considered as a major importance in NLP applications, especially in parsing,anaphora, information retrieval, question answering system, information extraction, named entity recognition and aspect-based sentiment analysis. This article was investigating the application of text chunking for Afaan Oromo language (AOL), using Conditional random fields and Support vector machine. AOL is an East Cushitic language; family of the Afro-Asiatic language and it is the most widely spoken language in Ethiopia. In our Experiment, we considered three different scenarios, such as word, part of speech tags of tokens.In the first scenario, features set which are two words from left and two words from right of the current word without Part of speech tagging(POS)tags were used. In the second scenario, we used all features within corresponding POS tag of each tokens as a feature. In the third scenario, we used one word from left and oneword from right of the current word within corresponding POS tag of each tokens as a feature. The better result of Conditional random field (CRF) and Support vector machine (SVM) methods are 87.4% and 86% is achieved respectively. This research indicated that a conditional random field (CRFs) was more applicable to Afaan Oromo (AO) text chunking than SVM.*

***Keywords**: Afaan Oromo (AO), Text chunking, shallow parsing, Conditional random fields (CRF), Support vector machine (SVM)*

Date of Submission: 04-06-2020 Date of Acceptance: 20-06-2020

I. Background of the Study

Natural language processing (NLP) is a sub field of Artificial Intelligence which deals with the ability of computer systems to analyse and synthesize spoken and written languages as human beings.NLP is concerned with the progress of computational models of human language processing. It is an interdisciplinary research area at the border between linguistics and artificial intelligence aiming at developing computer programs by using natural language text or speech according to the linguistics rules **[1]**. NLP is making computers to perform useful tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech **[2]**.NLP researchers aim to gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand, manipulate natural languages to perform the desired tasks**[3]**.

NLP objectives is designing and building software/application that will analyse, understand, and generate human languages, so that eventually humans will be able to communicate with the machines using natural language like AO. Some of applications that make use of NLP to allow users interact with computer systems using natural languages are machine translation, information extraction and retrieval using natural language, text-to-speech synthesis, automatic written text recognition, grammar checking, named entity recognition, sentiment analysis, parsing, chunking and Part – Of - Speech (POS) tagging.

The researchers of NLP use natural language technologies to develop computer applications for natural language. From this NLP investigation area or application, our work is mainly focussing on Text chunking or shallow parsing.Chunking or shallow parsing is NLPapplication that attempts to provide process of dividing sentences into series of words that together constitute a grammatical unit (mostly either noun or verb, or postposition phrase). It is identifying groups of contiguous words that depend on head of phrase in the sentences or texts. The term phrase is a group or combination of two or more words in the sentence.

AO is one of a major African languagethatis widely spoken in Ethiopia. The native speakers of the language are the Oromo people; they are the largest ethnic group in Ethiopia. It has around 40 million speakers, 34% of the total population of the country, native speakers and the most populous language of Ethiopia [4, 5, and 6].According to Tabor Wami[6]it is also the third most widely spoken language in Africa next to Arabic and Hausa languages. Specially, it is first widely spoken and used in most parts of Ethiopia and also some parts of other neighbour countries like Kenya, Tanzania, Djibouti, Sudan and Somalia[7]. research works on NLP like Text chunking are scarce. Thus, this research work is concerned with AO Text chunking.

Developing chunking or shallow parsing is challenging task especially for languages having low resource and complex linguistic structure like AO. Small amount of resources highly impacts the accuracy of the system. Complex structure of languages also needs design of important features and the best combination from these features. AO as one of low resourced and morphologically rich languages shares the above challenges in the designing NLP applications. The NLP applications like Text chunking can be broadly classified into rule based (linguistic approach), machine learning approaches and hybrid approach. Rule based approaches are based on handcrafted rules by language experts. Machine learning approaches are investigating how computers can learn (or improve their performance) based on data. A main research area is for computer programs to automatically learn to recognize complex patterns andmake intelligent decisions based on data[8].

There are works that has been done under different languages for text chunking using different approaches like rule based, machine learning and hybrid. But adopting directly the approach or tool developed for different language is difficult for AO. Naturally, Natural languages have own unique grammar, syntax and feature of word. Previous some AO NLPs like information extraction, information retrieval, named entity recognition, POS tagging, sentiment analysis and anaphora were developed for solving the natural language processing area. These applications were notidentified syntactically correlated parts of words in a text. Especially, In AOL, some words comprise two words to represent single words are syntactically correlated parts of word. As example: 'mana barumsaa'/School. This comprises words of AO is representing single word school in English. Therefore, applications used to identify syntacticallycorrelatedparts of words in a text were needed for AOLwhich was the objective of this research.

1.1. Statement of the problem

Text chunkingcan be useful for developing parsing,anaphora, information retrieval, question answering system, information extraction, named entity recognition and aspect-based sentiment analysis. The term parsing is an NLP technique that attempts to provide some understanding of the structure of a sentence into each word. Parsing is hierarchically categorized under syntax, and is used to generate valid parse tree for a sentence given grammar and lexical rules. Parsing uses recursive phrase structure grammars and arbitrary-depth trees.Parsing has problems with robustness, given the difficulty in getting broad coverage and in resolving ambiguity. It is also relatively inefficient: the time taken to parse a sentence grows with the cube of the length of the sentence, while the time taken to chunk a sentence only grows linearly. For these, use chunking text is easier or more efficient [2]. Also, in the case of high ambiguity of the natural language to exact parsing of the text may become very complex. In these cases, chunking can be used as component to partially resolve these ambiguities. It is also widely used as an intermediate step to parsing for the purpose of improving the performance of the parser[11]. To develop automatic parser for AO, chunking for AO is important to minimize time and effort [9, 10].

In the information retrieval system, chunking can be used to retrieve the data from the documents depending on the chunks rather than the words [12]. In the developing anaphora application, chunking is helpful forthat antecedent that comprises two or more words[13]. In the developing question answering system, query generation is a component of question answering system. In AOL, query is either single word or comprises two words. For two comprise words and syntactical correlated words, using chunk in the pre-processing was improve the quality of the system. Chunking is also helpful in the developing information extraction and named entity recognition system for that name of thing comprises from more than one word. In the developing Aspect based sentiment analysis, chunk deals with extract aspect or entity documents.

Therefore, text chunker application which recognize the types of phrases in to noun phrase, adjectival phrase, adverbial phrase, Ad positionalphrase and verb phrase are investigated. But, to the best knowledge of the researcher, text chunker has not been studied for AO yet. To this end, this study tries to answer the following research questions:

- ➢ What are the feature sets that will allow the CRFs and SVM to perform better result?
- ➢ From CRFs and SVM, which method fit chunking system from AO text?

1.2. Objectives

1.2.1. General objective

The general objective of this study was to investigate AO chunking using conditional random fields and support vector machines.

1.2.2. Specific objectives

The specific objectives of this research work were: -

- ➢ To prepare dataset for training and testing purposes.

- To identify feature set used for developing chunking using CRFs and SVM.
- To build a prototypefor chunking fromAO text.
- Evaluate the performance of the prototypemodel.
- To forward conclusion and recommendations based on result of this research.

1.3. Methodology

1.3.1. Data collection and preparation

AO does not have publicly available annotated corpus text for NLP tasks. To investigate this research, we performed manually three tasks: collected 1,000 sentences, POS tagged and chunk tagged.

1.3.2. Development tool

We used anaconda platform which includes a wide range of state-of-the-art machine learning algorithms for supervised and unsupervised problems. For this study we used supervised machine learning algorithms of Scikit-learn and sklearn-crfsuite tool. From Scikit-learn, we applied Support vector machine algorithms (LinearSVC); and we applied sklearn-crfsuite (CRFs) algorithms.

1.3.3. Performance Analysis

Performance of the proposed system was analysed using an analyser metrics of Scikit-learn and sklearn-crfsuite tool.

1.4. Scope and limitation of the study

This study was conducted to develop AOchunking system. The prepared dataset was including both purpose-based sentence (declarative, interrogative, imperative and exclamatory) and structure-based sentences (simple, compound, complex and compound-complex) of AO. This dataset was manual annotated dataset. The case of used the manual dataset was no any annotated corpus of AO like Treebank and CoNLL2000 for English. The annotated dataset was consisted 1,000 sentences with 6624 tokens and 24 POS tag set.

1.5. Application of Results

Chunking is one of natural language processing application which useful in the different NLP applications. Some of NLP applications which use chunking are parsing,anaphora, information retrieval, question answering system, information extraction, named entity recognition and aspect-based sentiment analysis.

1.6. Parsing

Parsing is hierarchically categorized under syntax, and is used to generate valid parse tree for a sentence given grammar and lexical rules. Chunking is the task of identifying and segmenting phrases from text and where parsing constructs deeply nested structures for the result of chunking. It is considered as an intermediate step towards full parsing.

1.7. Information retrieval

Information retrieval is about returning the information that is relevant for a specific query or field of interest. In information retrieval, retrieving relevant data from the corpus or documents is depends on query. If this query is phrase level, the retrieved result is more relevant [11]. As an example: '*innihoolaaadii bite'*. (He bought white sheep). [White sheep] or [*hoolaaadii*] is a phrase that is adjectival phrase. If we take this example to know about: 'what color of sheep bought by him?' For this question, the retrieve information by phrase query [white sheep] is rather relevant than the word query [sheep].

1.8. Anaphora

Anaphora means a word or phrase that refers to an entity that is mentioned previouslyand this word or phrase that it refers to is called its antecedent. Chunking is helpful forthat antecedent that comprises two or more words[13]. For example, let us consider the name school in Afaan Oromo "mana barumsaa" which is composed of two words.if we take the output of POS tagger "mana "is tagged as NN and "Barumsaa" is also tagged as NN but both words follow each other and also belong to the same thing. For solve like this problem, chunking is important.

1.9. Question answering system

A question answering is a task that aims to automatically give answers to questions described in natural language. It allows users to have exact answer rather than having list of potentially relevant documents. In the developing question answering system, query generation is a component of question answering system. In AOL, query is either single word or comprises two words. For two comprise words and syntactical correlated words, using chunk in the pre-processing was improve the quality of the system.

1.10. Information extraction

Information extraction is the way to extract information from a text in the form of text strings and processed text strings which are placedinto slots labelled to indicate the kind of information that can fill them. To extract information, information extraction flows some procedure such as:first, the raw text of the document is split into sentences and each sentence is further subdivided into words which is tokenization. Next, each segmented sentence is tagged with POS tags, which will prove very helpful in the next step, named entity detection. After named entity detected, transfer into relation detection, the sentence could chunk in the between named entity detection component and relation detection component.

1.11. Named Entity Recognition

Named entity recognition is a system which identifies the entity or the names ofpeople, places, organizations, dates, amounts of money and other entity from text. The entity or the names of things represent single or more than one word in the text. Chunking is helpful for that name of thing comprises from more than one word.

1.12. Aspect based sentiment analysis

Aspect based sentiment analysis is an aspect/feature level in which identifying and extracting the feature of the objects that have been commented on by the opinion holder and determining whether the opinion is positive, negative and neutral. Chunking used to extract a thing or entity as aspect from comment text.

II. Basic Concept of Chunking

As discussed before, identifying phrases in sentences is useful for exploring different NLP applications. Sentence is divided into the phrases based on head of phrase in that sentence. Human being who is expert of a language is easy to identify the phrase in the sentences. Using rule of expert, there is also NLP concepts to identify phrases in sentences. The process of determining or identifying phrases in sentence using NLP concepts is known as chunking (shallow parsing).Chunking or shallow parsing is a natural language processing application that is the way to identify syntactically correlated parts of words in a sentence. Chunking is the process of dividing the sentence into chunks. Chunks are the non-overlapping structure in a sentence. Chunking separates and segments a sentence into its sub constituents, such as noun phrase, verb phrase, adverb phrase, adjective phrase, adverb phrase and adjective positionalphrases. Shallow parsing or chunking is the alternative for full parsing in developing NLP applications [14].A chunker finds contiguous, non-overlapping spans of related tokens and groups them together into chunks. Chunkers often operate on tagged texts, and use the tags to make chunking decisions. The term tagged texts or part of speech taggingis the process of marking up a word in a text (corpus) as corresponding to a particular part of speech. After part of speech tagging is identified for word,then chunker chunks the tagged text into non-overlapping group that are noun phrase, verb phrase, adverb phrase or the others phrase.Chunk a text into correlated group of words or non-overlapping group is based on head of phrase in the sentence. Head of phrase means phrase which explain the content or the information about the sentence. For example: ***innigara mana deeme***. (He went to home). From this example, we learn the direction he moved ([***gara mana]*** (**'to home'**)). This phrase shows the direction he went. The phrases which show direction is called ad positionalphrase in AO. So, the head of phrase in sentences is ad positionalphrase that is [***gara mana].***

In AOL, there are five types of phrases such as noun phrase, verb phrase, ad positionalphrase, adverbial phrase and adjectival phrase. Based on this component, we proposed five type of chunk in AO. These are chunk of noun phrase, verb phrase, ad positionalphrase, adverbial phrase and adjectival phrase. Approaches for Chunking.There are different approaches to chunk the sentences or text of natural language. It can be categorized into rule-based approaches, machine learning approaches and hybrid.

1.13. Rule based approach

Rule based approach is a set of static rules which is created by language experts. Rule based approach uses regular expressions rule to match text within given pattern. The advantages of rule-based approach are extremely simple to implement and it does not require training corpus. However, there is a shortage of rule-based techniques in developing natural language processing applications. Such as: Rules are not sufficiently complete; they cannot predict the new data and exceptions that occur naturally in language (work for specifically applied structure and not predict the others) [7]. The other drawback of rule base is on generating new rule. Generating new rule is complex because naturally the human language is complex. The percentage of accuracy in rule-based approach is less than machine learning approach if the dataset has large size [15].

1.14. Machine learning approaches

Machine learning investigates how computers can learn (or improve their performance) based on given data. Machine learning approacheslearn based on statistical models to make predictions for new data depending on learning the given data. Machine learning approaches are Supervised, Unsupervised and Semi-supervised. Supervised learning approach is the type of machine learning which learn from labelled train data and predict the new data based on the trained dataset whereas; the unsupervised learning process is unsupervised since the input examples are not class labelled. Semi-supervised learning is a class of machine learning techniques that make use of both labelled and unlabelled examples when learning a model. In one approach, labelled examples are used to learn class models and unlabelled examples are used to refine the boundaries between classes. From this machine learning algorithms, we focus on supervised machine learning algorithms that is more applicable to text chunking area especially for under resourced language like AO. From those machines learning, we focused on supervised machine learning which is more relevant to developing text chunking applications. Some of the supervised machines learning approaches which are related to our investigation that explored for different language are: Hidden Markov model,Maximum entropy model,Naïve Bayesian,Memory-Based Learning, neural networks, support vector machine and Conditional random fields**[17]**.

Hidden Markov model (HMM) is a model for a statisticalprocess that generates sequential data. In HMM, there are observed events and hidden events in developing model. Observed event is the token – l which we input and hidden event is labelled part. Observation is observed based on the hidden event or variable. The trainer trains such models based on data, and / or use a trained model to try and infer the hidden states, or just get the probabilities of different observation sequences.The states or event processed in hidden Markov model are discrete number. The probability distribution of future states must depend only on the present state and be completely independent of past states. The states before the current state have no impact on the future except via the current state. For instance, if it predicts tomorrow weather you could examine today weather but you weren't allowed to look at yesterday weather.

Maximum entropy model (MEM) is also supervised machine learning algorithm used in sequential classification like HMMs.ButMEM solves the problem of multiple feature representation and long-term dependency issue that occurs in HMM. It has increased the recall and greater precision than Hidden Markov Model**[16]**.Memory-Based Learning (MBL) is supervised learning approach that classifying data based on their similarity to data that they have seen earlier. Memory-based learning algorithm is constructing a classifier for a task by storing a set of examples **[12]**.

A neural network is also an algorithm that designed for both supervised and unsupervised data.Neural network is process information in a similar way of human brain. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. Neural network is containing cyclic connections, which enable it to learn the temporal dynamics of sequential information. In the training process, neural network uses most data to improve best accuracy of system **[19].**Naïve Bayesian is one of the generative models which are the types of supervised machine learning. Naïve Bayesian is mostly used in text classification of natural language processing. It belongs to the group of graphical models which is used to model conditional independence between random variables. The Bayesian model is based on the use of Bayes law of probability which uses the inverse of conditional probability[39]. Next, we discussed briefly the supervised machine learning algorithms that we have used in our experiment which are support vector machine (SVM) and Conditional random fields (CRF).

1.15. Support vector machine (SVM)

A support vector machine (SVM) is a type of supervised machine learning classification algorithm. SVM was first introduced by Vladimir Vapkin and his colleagues [18]. According to **[18]** SVM are well-known for their good generalization performance and have been applied to many recognition problems, including chunking, named entity recognition, parsing and text categorization to mention some**[18].**Also stated that there are theoretical and empirical results that indicate the good performance of SVMs' ability to generalize in a high dimensional feature space without over-fitting the training data than others distance or similarity based learning algorithms such as k-nearest neighbour (KNN) or decision tree. SVM is also differed from the other classification algorithms in the way that it chooses the decision boundary that maximizes the distance from the nearest data points of all the classes. An SVM doesn't merely find a decision boundary; it finds the most optimal decision boundary. The most optimal decision boundary is the one which has maximum margin from the nearest points of all the classes. The nearest points from the decision boundary that maximize the distance between the decision boundary and the points are called support vectors as seen in below figure. The decision boundary in case of support vector machines is called the maximum margin classifier, or the maximum margin hyper plane**[19]**.

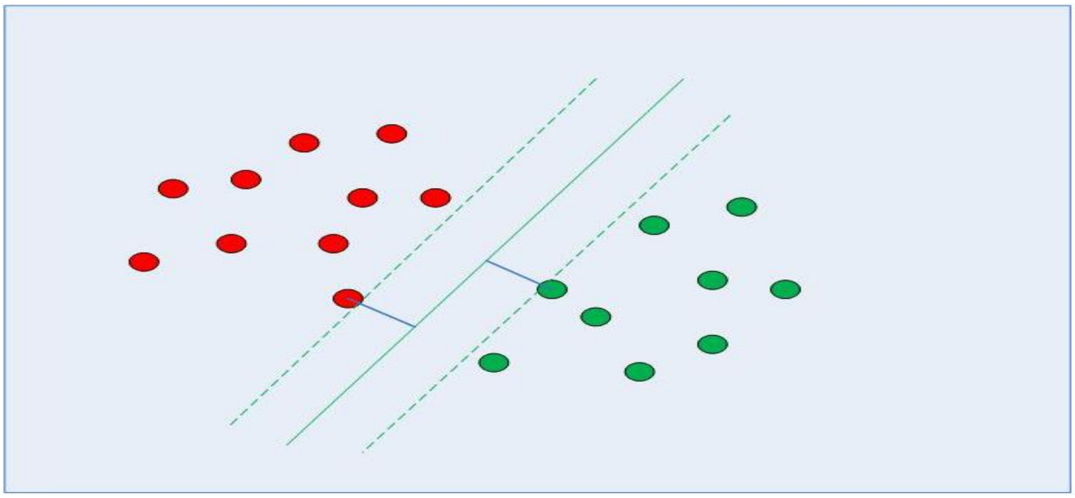

Figure 1: Decision boundaries with support vectors

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. Support vectors data points are more relevant to the construction of the classifier. The term hyperplane is a decision plane which separates between a set of objects having different class memberships. The others term is margin. Margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin**[18]**.In their basic form shown in Figure 1, SVM construct the hyperplane in input space that correctly separates the example data into two classes. This hyperplane can be used to make the prediction of class for unseen data. The hyperplanes always exist for the linearly separable data**[18]**. The aim of hyper planes in the training phrase of the SVM is to plot the vectors in n-dimensional hyperspace and draw a hyper plane as evenly as possible to separate points from the two categories. The formula description of support vector machine in separating points using hyper planes has described by **[19]**. As an example: in the classification when xi as a feature and $y_i$is a label class either 1 or 0 (Whether y_i is positive or negative indicates the category for the vector i). Assume, this hyper plane has normal vector w. then the hyper plane can be written as the points x satisfying:

$$W.X - b = 0 \qquad (1)$$

Where $\frac{b}{\|w\|}$is the offset of the hyperplane from the origin along w. This hyper plane is chosen so as to maximize the margin between the points representing the two categories. Imagine two hyperplanes lying at the 'border' of two regions in each of which there are only points of either category. These two hyperplanes are perpendicular to w and cut through the outermost training data points in their respective regions. Two such planes can be seen illustrated as dashed lines in above figure. Wanting to maximize the margin between the points representing the two categories is the same thing as wanting to keep these two hyperplanes as far apart as possible The training data points which end up on the dashed lines in above figure are called support vectors, hence the name Support Vector Machine. The hyperplanes can be described by the equations.

$$W.X - b = 1 \qquad (2)$$

$$W.X - b = -1 \qquad (3)$$

The distance between the two is$\frac{2}{\|W\|}$. Since the SVM wants to maximize the margin, we need to minimize $\|W\|$. It also does not want to extend the margin indefinitely, since it does not want training data points within the margin. Thus, the following constraints are added to the problem:

$w.x_i \geq \ 1$	For$x_i$in the first category, and
$w.x_i \geq -1$	For$x_i$in the second category

This problem can be rewritten asthe optimization problem of minimizing ||w|| subject to

$$(w.x_i - b)y_i \geq 1 \ for \ (1 \leq i \leq n)$$

Now replaces$\|W\|$ with$\frac{2}{\|W\|}$, and use Lagrange Multipliers and then rewrite this optimization problem into the following Quadratic Optimization problem:

$$\min_{w,b} \max_{\alpha_i \geq 0} \left\{ \frac{\|W\|^2}{2} - \sum_{i=1}^{n} \alpha_i \big((w.x_i - b)y_i - 1 \big) \right\}, \alpha_i \geq 0 \qquad (4)$$

Whereα_i they are Lagrange's Multipliers. Data sets which are possible to divide in two are called linearly separable. Depending on how the data is arranged, this may not be possible. It is, however, possible to use an alternative model involving a soft margin. The soft margin model allows for a minimal number of mislabelled examples. This is done by introducing slack variables i for each training data vectorx_i The function to be minimized, $\frac{\|W\|^2}{2}$ are modified by adding a term representing the slack variables. This can be done in several ways, but a common way is to introduce a linear function, so that the problem is to

Minimize $\frac{\|W\|}{2} + C\sum_{i=1}^{n} d_i$ (5)

For some constant C, to this minimization, the following modified constraint is added:

$(w.x_i - b)y_i \geq 1 \ for \ (1 \leq i \leq n)$ (6)

By using Lagrange Multipliers as before, the problem can be rewritten as:

$\min_{w,d_i,b} \max_{\alpha_i,\beta_i} \left\{\frac{\|W\|^2}{2} + C\sum_{i=1}^{n} d_i - \sum_{i=1}^{n} \alpha_i\big((w.x_i - b)y_i - 1 + d_i\big)\right\} - \beta_i d_i \geq 0 \ for \ b \geq 0$ (7)

To get rid of the slack variable we rewrite the problem into its dual form:

$\max_{\alpha} \left\{\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{m} \alpha_i\alpha_j y_i y_j x_i x_j\right\}$ (8)

Subject to constraint

$0 \leq \alpha_i \leq C \quad for \ (1 \leq i \leq n)$ And also$\sum_{i=1}^{n} \alpha_i \, y_i = 0 \ for \ (1 \leq i \leq n)$

The above formal description on SVM applies only to those problems which are linearly separable. In a non-linear classifier, the input vectors xi are transformed as to lie in an infinitely dimensional Hilbert Space where it is always possible to linearly separate the two data categories.Hence SVM is a binary classifier. The SVM described in the above formula explained by **[19]** is used for binary classification and which classify data in two categories. But Text chunking is a multi-class classification problem since in natural language there are more than two label categories. For multi class problem, support vector machine can solve multi classification problem using different method. Some method in SVM that used to solve multi classification problem are: one-against-one and one-against-rest**[20]**. In the one-against-one approach, instead of trying to distinguish one class from all the others, they seek to distinguish one class from another one. In the one-against-rest binarization of problem, SVM is trained for each class in order to distinguish that class and the rest. When we used this both method of SVM, one-against-rest method are more suitable for practical than one-against-one. The one-against-all method has the added advantage of being already available in sclera (Linear SVC library), so it should probably be our default choice.

1.16. Conditional Random Fields

Conditional random fields (CRF) is machine learning approach that uses discriminative learning for automatically learning the rules from the annotated training data and produces model which is applied to the test data specially for sequence classification [44]. Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting structured data **[22]**. The underlying idea is that of defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences.

CRFs is probabilistic model for computing the probability $P(\vec{y}|\,\vec{x})$ of a possible output sequence $\vec{y}$ = $(y_1, \ldots, y_n)$ as $\sum y$ given the input sequence $\vec{x}$= $(x_1, \ldots, x_n)$ as $\sum x$that is called observation. y is the set of all possible output category sequence and x, it is the set of observation sequence: Under this section a special form of CRFs which is linear chain CRFs is discussed in detail that is used in our case. Linear chain CRFs is a special form of CRFs, which is structured as a linear chain that models the output variables as a sequence. Linear-chain CRFs can be formulated as **[24]**as given below:

$P\,\vec{\lambda}\,(\vec{y} \mid \vec{x}) = \frac{1}{z\,\vec{\lambda}(\vec{x})} exp\big(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i \, f_i\big(y_{j-1}y_j, \vec{x}, j\big)\big)$ (9)

Where x it is a data sequence that is observation, y it is Type equation here.a category sequence of data, n indicates the length of sentence, m indicates number of feature templates,$\lambda_i$represents the weights assigned to the different features in the training phase and $z\,\vec{\lambda}\,(x)$, it is a normalization factor that makes the probability in the range$[0, \quad 1]$, This equation (9) can be expressed as:

$z\,\vec{\lambda}\,(\vec{x}) = \sum_{\vec{y} \in y} exp\left(\big(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i \, f_i\big(y_{j-1}y_j, \vec{x}, j\big)\big)\right)$ (10)

The next sub sections will respectively discuss how a linear chain CRF model is used in the developing our model.

1.17. Feature extraction function

The basic part developing natural language processing application using machine learning is identifying a feature extraction. Identifying features is the most important process for machine-learning approaches because a feature design greatly affects the labeling accuracy **[14]**. The feature extraction is the components of CRFs. In our special case of Linear-chain CRFs, the general form of a feature extraction function is $f_i\big(y_{j-1}y_j, \vec{x}, j\big)$, which looks at a pair of adjacent states y_{j-1}, y_j the whole input sequence $\vec{x}$, and where we are in the sequence(j). These are arbitrary functions that produce a real value. As an example: we can define a simple feature function which produces binary values

$$f_i(y_{j-1}y_j,\overrightarrow{y},j) = \sum_{0}^{1} y_{j-1}y_j = Noun\ and\ x_j = \begin{cases} jimma \\ otherwise\ 0 \end{cases}$$

That is, "If they^{th}word is $jimma$ and having a tag noun, and then$f_i = \begin{cases} 1 \\ otherwise\ 0 \end{cases}$

The usage of this feature depends on its corresponding weightλ_1. If $\lambda_1 > 0$, whether f_1 is active, it increases the probability of tag sequence$y_1: n(\overrightarrow{y})$This is another way of saying the CRFs model should prefer the tag Noun for the word $jimma$. Otherwise $\lambda_1 < 0$, the CRF model will try to avoid the tag Noun for $jimma$. In this case the value of λ_1 will be learned from the corpus. Addressed two problems of Linear chain CRFS:

Problem – 1:

Given observation $\boldsymbol{x}$ **and CRF M,** how do find the model probability fitting label sequence$\overrightarrow{y}$?

Note that this this type of problem is the most common application of a conditional random field to find a label sequence for an observation.

Problem – 2:

Given label sequence $\boldsymbol{y}$ **and observation sequence x.** How do find parameters of a CRF, M to maximize $P\overrightarrow{\lambda}(\overrightarrow{y}\ |\overrightarrow{x}, M)$?

This problem is questioned of how do train to adjust the parameters of M, which are especially the feature weight $\overrightarrow{\lambda}$?

1.18. CRFs Model Training

For all types of CRFs, the maximum-likelihood method can be applied for parameter estimation. That means, training the model is done by maximizing the log-likelihood Z on the training data T: We can find in the following manner:

$$L(T) = \textstyle\sum_{(\overrightarrow{x},\ \overrightarrow{y})\in T} \log[P(\overrightarrow{y}|\ \overrightarrow{x})] = \sum_{(\overrightarrow{x},\ \overrightarrow{y})\in T} \left[\log\left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{j-1}y_j,\overrightarrow{x},j)\right)}{\sum_{\overrightarrow{y'}\in y}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y'_{j-1}y'_j,\overrightarrow{x},j)\right)}\right)\right] \quad (11)$$

To avoid over fitting the likely – hood is penalized with the term: $-\sum_{i=1}^{n}\frac{\lambda_i^2}{2\sigma^2}$

The parameterσ^2 models the trade – off between fitting exactly the observed feature frequencies and the squared norm of the weight vector. The smaller the values are, the smaller the weights are forced to be, so that the chance that few high weights dominate is reduced. For the purpose of avoiding over fitting, the likely – hood function $L(T)$, rewrite as follows:

$$L(T) = \textstyle\sum_{(\overrightarrow{x},\overrightarrow{y})\in T} \left[\log\left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{j-1}y_j,\overrightarrow{x},j)\right)}{\sum_{\overrightarrow{y'}\in y}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y'_{j-1}y'_j,\overrightarrow{x},j)\right)}\right)\right] - \sum_{i=1}^{n}\frac{\lambda_i^2}{2\sigma^2} \quad (12)$$

$$\Rightarrow \frac{\partial}{\partial\lambda_k}[L(T)] = \underbrace{\textstyle\sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{j-1}y_j,\overrightarrow{x},j)}_{A} - \underbrace{\frac{\sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\sum_{(\overrightarrow{x},\ \overrightarrow{y})\in T}\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{j-1}y_j,\overrightarrow{x},j)}{z\overrightarrow{\lambda}(\overrightarrow{x})}}_{B} - \underbrace{\textstyle\sum_{i=1}^{n}\frac{\lambda_i^2}{2\sigma^2}}_{C} \quad (13)$$

Since$\frac{\partial}{\partial\lambda_k}[L(T)]$ with respect λ_k, they are computed separately for the parts A, B and C.

The derivation of part A is given by:

$$\frac{\partial}{\partial\lambda_k}\left[\textstyle\sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{j-1}y_j,\overrightarrow{x},j)\right] = \sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\sum_{j=1}^{n} f_k(y_{j-1}y_j,j) \quad (14)$$

The derivation part of B which corresponds to the normalization is given by:

$$\frac{\partial}{\partial\lambda_k}\sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\log z\overrightarrow{\lambda}(\overrightarrow{x}) = \sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\frac{1}{z\overrightarrow{\lambda}(\overrightarrow{x})}\frac{\partial z\overrightarrow{\lambda}(\overrightarrow{x})}{\partial\lambda_k}$$

$$\Rightarrow \sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\frac{1}{z\overrightarrow{\lambda}(\overrightarrow{x})}\frac{\partial z\overrightarrow{\lambda}(\overrightarrow{x})}{\partial\lambda_k} = \sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\frac{1}{z\overrightarrow{\lambda}(\overrightarrow{x})}\sum_{\overrightarrow{y'}\in y}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y'_{j-1}y'_j,\overrightarrow{x},j)\right)\sum_{j=1}^{n} f_i(y'_{j-1}y'_j,\overrightarrow{x},j)$$

$$\Rightarrow \textstyle\sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\frac{1}{z\overrightarrow{\lambda}(\overrightarrow{x})}\frac{\partial z\overrightarrow{\lambda}(\overrightarrow{x})}{\partial\lambda_k} = \sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\sum_{\overrightarrow{y'}\in y}\frac{\frac{1}{z\overrightarrow{\lambda}(\overrightarrow{x})}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y'_{j-1}y'_j,\overrightarrow{x},j)\right)}{P\overrightarrow{\lambda}(\overrightarrow{y}|\overrightarrow{x})} \quad \text{(By equation (9))}$$

$$\Rightarrow \textstyle\sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\frac{1}{z\overrightarrow{\lambda}(\overrightarrow{x})}\frac{\partial z\overrightarrow{\lambda}(\overrightarrow{x})}{\partial\lambda_k} = \sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\sum_{\overrightarrow{y'}\in y} P\overrightarrow{\lambda}(\overrightarrow{y'}\ |\ \overrightarrow{x})\sum_{j=1}^{n} f_k(y'_{j-1},y'_j,\overrightarrow{x},j) \quad (15)$$

Since C is the derivation of the penalty term and is given by:

$$\frac{\partial}{\partial\lambda_k}\left(-\textstyle\sum_{i=1}^{n}\frac{\lambda_i^2}{2\sigma^2}\right) = -\frac{2\lambda_k}{2\sigma^2} = -\frac{\lambda_k}{\sigma^2} \quad (16)$$

Since Equn. (14) is the derivation part of A which is the expected value under the empirical distribution of featuref_i: That us

$$\overleftarrow{E}(f_i) = \textstyle\sum_{(\overrightarrow{x},\overrightarrow{y})\in T}\sum_{j=1}^{n}\sum_{i=1}^{m} f_i(y_{j-1}y_j,\overrightarrow{x},j) \quad (17)$$

By Equn. (15), the derivation part of B, is the expectation under the model distribution:

$E(f_i) = \sum_{(\vec{x},\vec{y})\in T} \sum_{\vec{y'}\in y} P\,\vec{\lambda}\,(\vec{y'} \mid \vec{x}) \sum_{j=1}^{n} f_k(y'_{j-1}, y'_j, \vec{x}, j)$ (18)

$\Rightarrow \frac{\partial}{\partial\lambda_k}[L(T)]$, it can also interpret as:

$\frac{\partial}{\partial\lambda_k}[L(T)] = \breve{E}\,(f_i) - E(f_i) - \frac{\lambda_k}{\sigma^2}$ (19)

To get the maximum by the approximation of the first derivation,

$\breve{E}\,(f_i) - E(f_i) - \frac{\lambda_k}{\sigma^2} = 0$ (20)

Now, to calculate each weighting valueλ_k for each featuref_k easily and also computing $\breve{E}\,(f_i)$ is easily done by Counting how often each feature occurs in the training data. Computing $E(f_i)$ is directly impractical because of the high number of possible tag sequence(y). In CRF, sequences of output variables lead to enormous combinatorial complexity. Thus, a dynamic programming approach is applied, known as the Forward – Backward Algorithm which is beyond the scope of this research

III. Text Chunking for AO

As far as the knowledge of the researchers concerns there were no system or work had been proposed in the past for Text chunking AOL. For, AO our research is the first for chunking AO text and it became the initial or starting point for other researcher who was interested on these areas. The summary of articles, methodology, dataset used, features, language and accuracy obtained are shown in the below table.Text chunking systems were summarized by using different approach in the below table. The rule-based approach is good result in very small data size and it is dependent on linguistic experts. Machine learning approach is better performance than rule-based approach. However, in Machine learning approaches, a pre-tagged or an annotated text is required to train the system. The accuracy of Machine learning chunker directly depends on the size of training data set. As the size of training data increases, the accuracy also increases. The calculated result of machine learning is also different.From discussed approaches, CRFs and SVM achieved better performance than others machine learning in the reviewed papers. Therefore, in this research we applied conditional random fields and SVM machine learning algorithms for AO Text chunking. Text chunking involves the identification and divides sentences into correlated group of words such as noun phrases, verb phrases, adjective phrases, adverb phrases and ad positional phrases. It falls within the category of sequential labelling like POS tagging, word segmentation, named entity recognition and information extraction. the implementation of CRFs and SVM based chunking system for AO will be discussed as follows: the first section talks about data collection and preparation. Second section describes over view of the chunking boundary format; third section was architecture of AO chunking.

Articles	**Methods**	**Data set**	**Features**	**Language**	**Performance %**
Kuang – hua Chen and Hsin – His Chen, 1993 [22]	**Statistical**	**Brow Corpus**	**Linguistic Rules**	**English**	**Accuracy 98**
Chakaraborty, 2016 [23]	**Rule Based**	**50 Documents**	**Linguistic Rules**	**English**	**Accuracy 84**
Kudo and Matsumoto, 2001[25]	**SVM**	**Penn Tree Bank**	**Lexical + POS tag**	**English**	**F – Score 94.22**
Sneha Asopa et.al. 2016 [23]	**Rule Based**	**500 Sentences**	**Linguistic Rules**	**Hindi**	**Accuracy 74.16**
Akshay Singh et.al. 2009 [15]	**HMM**	**2, 00, 000**	**Lexical**	**Hindi**	**Accuracy 91.7**
Koeling 2000 [16]	**MEM**	**2, 00, 000**	**Lexical**	**Hindi**	**Recall 91.86**
Guang - Lu et.al. 2010 [17]	**Naïve Bayes +Semantic Feature**	**Hownet Dictionary 67, 440 words**	**Lexical +POS**	**Chinese**	**Precision 93.04 Recall 91.86 F – Score 92.8**
Guang - Lu et.al. 2016 [26]	**MEMA**	**CPTB and MSRA**	**Lexical +POS**	**Chinese**	**F – 92.16 CPTB F – 91.02 MSRA**
Prathibha R. J. Padma M. C. 2017 [24]	**CRF**	**EMILLE (6, 000 Sentences)**	**Lexical +POS**	**Kanada0**	**Accuracy in Novels 92.77 and in stories 93.28**
Abeba I. 2013 [8]	**HMM + Rule**	**320 Sentences**	**Lexical +POS**	**Amharic**	**Accuracy 93.75**

Table: 1 Summary of Related work

1.19. Data collection and Data preparation

For this research, we prepared sentences manually by reading grammar book of AO and the others sentences were prepared by AO teachers in Bule Hora University and Jimma University in Ethiopia. This prepared includes both purpose-based sentences and structural based sentences of AO. The prepared sentences in this research were1,000sentences which include both purpose-based sentences and structural based sentences of AOL. The sentences that were collected by researcher and data collectors were not tagged. After collected the sentences, and were prepared of data set in the form machine learning can learn from. This preparation data set was assigning part of speech tagging for each token and identifying noun phrase boundary in sentences. Part of Speech Tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context i.e., its relationship with adjacent and related words in a phrase or sentence. In sentences, all words can be labelled with their POS tag. These tags denote the grammatical function of the word in the sentence. Some simple, but well-known POS tags are for instance nouns, verbs, adjectives, adverbs and determiners. POS tagging makes sentences easier to parse or chunking by a computer, and is therefore a pre-processing step frequently used in text-processing systems. Tag set used in this research reported in appendix-1.

1.20. Chunk boundary formatin AO

We have prepared at 30 linguistic rules that are used to identify the boundaries of chunker for AOL in the prepared dataset and they are listed below. Noun chunking is a chunk of correlated group of words and noun is the head this chunk.We have got at 12 linguistic rules that are used to identify the boundaries of noun chunker for AOL. Such as:

1. Noun and noun
2. Noun and adjective
3. Noun and Possessive Pronoun
4. Noun and number
5. Noun and determiners
6. Noun, adjective and determiners
7. Proper nouns with ('$-n$')
8. Personal pronoun
9. Noun, Possessive Pronoun and determiners
10. Noun, adjective and number
11. Noun and noun phrase
12. Noun phrase and determiners

Verb chunking is the verb group includes the main verb and auxiliary verbs. We have prepared 5 linguistic rules that used to identify the boundaries of verb chunk. Such as:

1. Adverb and verb
2. Verb and verb
3. Adverb, verb and verb
4. Noun, verb and verb
5. Noun, adverb and verb

Adjective Chunk: Adjective chunk consists of all adjectives including predicative together with noun chunk. However, adjectives appearing before a noun will be grouped together with the noun chunk. Therefore, we designed 4 linguistic rules that used to identify the boundaries of adjectivechunk. Such as:

1. Adjective and adjective
2. Adjective and determiners or adjective, adjective and determiners
3. Adjective and number
4. Adjective, number and determiners

Ad positional chunk is chunking a group of word when pre postposition acts as head. We prepared 5 linguistic rules that used to identify the boundaries of postpositionchunk. Such as:

1. Noun and ad positions, pre-/postposition and conjunctions
2. Noun phrase and ad positions, pre-/postposition and conjunctions
3. Ad positions, pre-/postposition and conjunctions and noun
4. Ad positions, pre-/postposition and conjunctions and noun phrase
5. Ad positions, pre-/postposition and conjunctions and postposition phrase or postposition and ad positions, pre-/postposition and conjunctions

Adverb chunk: a chunk that includes all adverbial phrases and adverb act as a head. We organized 4 linguistic rules that used to identify the boundaries of adverbchunk. Such as:

1. adverb and adverb
2. from different adverb
3. adverb with ('-tti')
4. adjective and postposition with ('-tti')

Over all, for this research, we prepared 30 linguistic rules to determine the chunk boundaries in AO sentences and we used when we prepared data set. After the phrase boundary lengthwas identified, the next step was labelling the chunk. The POS tag set helps to assign the label on chunk.We used the IOB notation model in which every word is to be tagged. Every chunk included each of this notation models, for instance noun chunk contains B-NP and I-NP and others are also like this. The number of IOB label which we used in this designed were 11 that including O notation which is outside of any chunk. As an example: "**Dallaanlooniikaleessaijaarame.**" This is Afaan Oromo simple sentence which was annotated in the table form for training.

Tokens	POS	Chunk tag
Dallaan	NNP	B-NP
Loonii	NNP	I-NP
Kaleessa	AD	B-VP
Ijaarame	VB	I-VP
.	PUNC	O

Table 2: Sample chunk annotated AO sentences for training model

1.21. Architecture of the system

This sub-section elaborates the overall architecture of the proposed system. The system has three phases the training phase, the testing phase and the prediction phase.

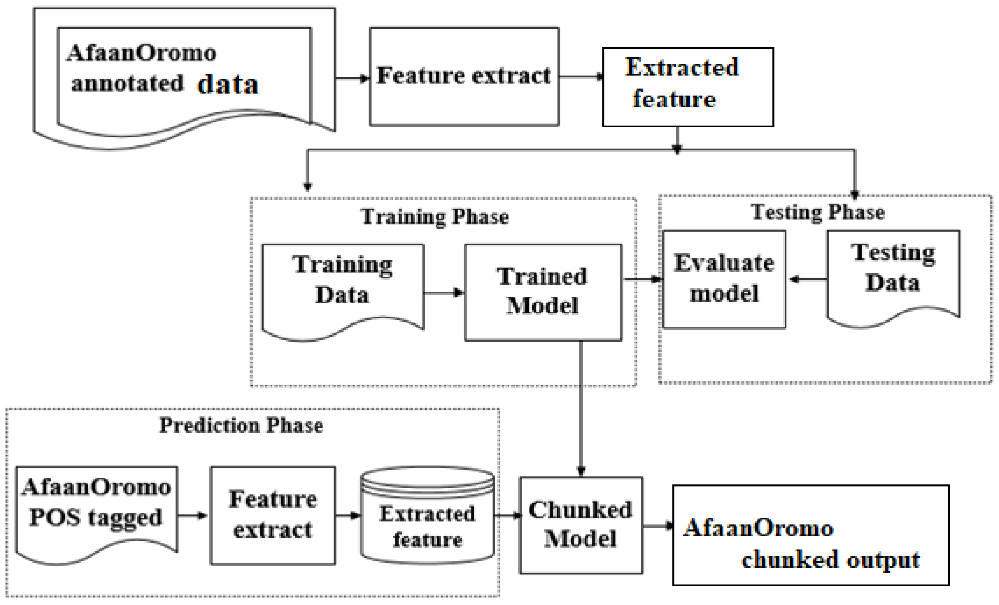

Figure 2: Proposed Architecture for AO Text chunking

3.3. AO annotated data

AO annotated data was prepared in the form of three excel columns that means the first columns is tokens, second columns is POS tag and third columns is chunk tags.

3.4. Feature Extraction

The basic part developing natural language processing application using machine learning is identifying a feature set. Identifying features is the most important process for machine-learning approaches because a feature design greatly affects the labelling accuracy [40]. Features indicate which information is used from the annotated corpus during the training stage. In this research we chose to use the word itself with POS tag annotated to build our features. These features express the characteristic of the word at position current word by using information from the context of words. Due toPOS tag of previous words and next words influence the chunk tag of current word we used the context window size is start from previous to previous word until next to next word of current words.

Feature	Definition
W_{i-2}	Previous to previous word
W_{i-1}	Previous word
W_i	Current word
W_{i+1}	Next word
W_{i+2}	Next to next word

POS_{i-2}	POS of previous to previous word
POS_{i-1}	POS of Previous word
POS_i	POS of current word
POS_{i+1}	POS of next word
POS_{i+2}	POS of next to next word

Table 3: Feature extraction window size in the proposed system

3.5. Training phase

The training dataset in Machine Learning is the dataset that is used to train the model for performing various actions. This is training dataset is the ongoing development process models learn with various algorithm to train the machine to work automatically. Shortly, the sample of data used to fit the model is known as training dataset. This training phase is for building the model. For this research, we used the conditional random field (CRF)(sklern-crfsuite) and SVM ($Linear\ SVC$) library to develop trained model. Trained model was output which we get after fitting training target of x with training target of y when target of x is the feature which contains words with POS and target of y is label part that is chunk classes.

3.6. Testing phase

*The dataset which is not in training dataset that is used to evaluate the training model is called testing dataset.*In our case, after trained model was stored, we used stored trained model to evaluate the testing data.

3.7. Prediction phases

Prediction phase is the process to input new data in the form of training feature and display the categorized data into labels classes. In the text chunking the input data is a text which tagged with part of speech. After trained model was stored and testing feature was evaluated, we can predict new inputted using this model. Prediction in this model is predicted when the inputted data was pass in the trained model.Sample input for chunker model and Output obtained from the developed chunker model is given below:

Input:

[('Tolaan', 'PNS'), ('gara', 'APC'), ('manaa', 'NN'), ('deemu', 'VB'), ('qaba', 'VB'), ('.', 'PUNC')]

Predicted Output:

[(('Tolaan', 'PNS'), 'B-NP'), (('gara', 'APC'), 'B-PP'), (('manaa', 'NN'), 'I-PP'), (('deemu', 'VB'), 'B-VP'), (('qaba', 'VB'), 'I-VP')]

IV. Experiments And Results

In this content we discusseddevelopment tools in this research and experimental results of text phrase chunking using SVM and CRFs.

4.1. Development tools

The development tools used in the process of this research are anaconda platform for python programmingand Jupiter notebook (for editor interface), Scikit learn and sklearn-crfsuitemachine learning library.

Anaconda is data science and machine learning platform for the Python and R programming languages. It is designed to make the process of creating and distributing projects simple, stable and reproducible across systems and is available on Linux and Windows. For this experiment, we used anaconda which is a Python based platform that contain major data science packages including pandas, scikit-learn and NumPy. Pandas are a software library written for the Python programming language for data manipulation and analysis. Pandas DataFrames make manipulating your data easy, from selecting or replacing columns and indices to reshaping your data. The other library is NumPywhich is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices).Scikit-learn is a Python machine learning library which includes a wide range of state-of-the-art machine learning algorithms for supervised and unsupervised problems. It focuses on bringing machine learning to non-specialists by providing a general-purpose high-level language. Scikit learn is amachine learning library for the Python programming language. For this research, we used support vector machine learning algorithms from sklearn tool. Support vector machine (SVM) has many classes library that is capable of performing binary classification and multi classification. Some of them are: SVC, $Nu\,SVC$ and $linear\,SVC$library. SVC and $Nu\,SVC$ are similar methods and use kernel parameters. On the other hand, $linear\ SVC$ is another implementation of Support Vector Classification for the case of a linear kernel, but $linear\ SVC$ does not accept keywordkernel. Chunking has multi-classification concept that means chunking category are more than twocategories, in our case there are three categories. In the multi classification, SVC and $Nu\,SVC$implement the "one-against-one" approach for multi-class classification. If K is the number of classes, then$K(K-1)/2$ classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decision function shape option allows transforming the results of the "one-against-one" classifiers to a decision function of shape (n −sample, K).$Linear\ SVC$ automatically uses the one-against-all strategy by default. You can also specify it explicitly by setting the multiclass parameter to over (one-vs-the-rest). Theoretically, the term 'One-against-one 'is an approach used to classify one class from others classes. In this approach, instead of trying to distinguish one class from all the others, we seek to distinguish one class from another one. As a result, we train one classifier per pair of classes, which leads to $K(K-1)/2$ classifiers for K classes. Each classifier is trained on a subset of the data and produces its own decision boundary. For this experiment, we used LinearSVC library of SVM which contains default approach for multi-class classification used in sklearn. Sklearn-crfsuite is a thin conditional random field suite or it is python-crfsuite wrapper which provides interface similar to scikit-learn in Anaconda platform by python programming. Sklearn-crfsuite is a library which has implementation of CRFs for labelling and segmenting sequence data.

4.2. Performance evaluation

Scikit learn and sklearn-crfsuite machine learning has standard evaluation metrics by itself. From evaluation metrics, we used classification report measurements of performance system. A classification report is used to measure the quality of predictions from a classification algorithm by counting how many predictions are True and how many are False.

4.3. Experimental Results

To do this research, the annotated dataset consisted of 1000 sentences with 6624 tokens. 75% of this dataset were used for training and the left were testing. Table 5 shows details of training and testing dataset.

Dataset	Total sentence	Total tokens
Training	750	4968
Test	250	1656

Table 4: Statistics of training and testing dataset

To do the experiment, 3 different scenarios were considered. Those scenarios were used in different feature extractions and in the same dataset which discards 25% of training dataset for test. For evaluating the performance resultis listed below accordingly.

In the first scenario, all features which all features which include two words from left and two words from right of the current word without POS tags were used.

Methods	Features used	Accuracy (%)
CRF	$W_{i-2}, W_{i-1}, W_i, W_{i+1}, W_{i+2}$	78
SVM		79

Table 5: Performance of the AO text chunking system in scenario 1

In the second scenario, we used all features which include two words from left and two words from right of the current word within corresponding POS tag of each tokens as a feature.

Methods	Features used	Accuracy (%)
CRF	$W_{i-2}, W_{i-1}, W_i, W_{i+1}, W_{i+2}$ $POS_{i-2}, POS_{i-1}, POS_i, POS_{i+1}, POS_{i+2}$	87.4
SVM		85

Table 6: Performance of the AO text chunking system in scenario 2

In the third scenario, we used one word from left and oneword from right of the current word within corresponding POS tag of each tokens as a feature.

Methods	Features used	Accuracy (%)
CRF	W_{i-1}, W_i, W_{i+1} $POS_{i-1}, POS_i, POS_{i+1}$	86.3
SVM		86

Table 7: Performance of the AO text chunking system in scenario 3

4.4. Discussion result of CRFs and SVM

The maximum average of CRFs results are accuracy 87.4% in second scenario.We saw increased window size of tokens and POS tag are important feature for AO text chunking performance. In this research, the majority of dataset was short lengths that means simple sentences plus exclamatory and interrogative sentences are almost in number and they are short lengths. The size of dataset used in this experiment was also very small. For these cases, we used window size of feature until $-2/+2$. For this small dataset which contains majority of short length, increased window size of feature sets more and more were more immutable on the results.

Our second approach in this experiment was support vector machine (SVM). In support vector machine,SVM is chooses the decision boundary that maximizes the distance from the nearest data points. Therefore, rather than increased/decreased window size of tokens and POS tag as feature, maximum the nearest data $(-1/+1)$ achieved best performance. In this is experiment, the better result of SVM achieved 86% accuracy in third scenario. SVM is chooses the decision boundary that maximizes the distance from the nearest data points.

Toanswer the first research question, the feature set achieved best result is two words from left and two words from right of the current word within corresponding POS tag of each tokens in second scenario. To answer the second research question, CRFs method fit text chunking system for AOL in this dataset than SVM. Finally, from this result, we understand, CRFs was more applicable for AO text chunking than SVM in this dataset. The lower performance could be due to the concept SVM is text classification and it is choosing the decision boundary that maximizes the distance from the nearest data points and the CRFs approach is sequence labelling classification algorithm. Text chunking is a sequential labelling problem which has concept of CRFs.

V. Conclusion Andfuture Work

5.1. Conclusion

Chunking is the process of identify syntactically correlated parts of words in a sentence. Chunking can be useful in parsing,anaphora, information retrieval, question answering system, information extraction, named entity recognition and aspect-based sentiment analysis. This research was conducted for AO text chunking using Conditional random fields (CRFs) and Support Vector Machine (SVM). To do this research, 1,000 sentences dataset (750 training and 250 testing) were used. This sentence was collected manually for this work. After collecting sentences, we were labelling POS tag and chunk tag. During the experiment, three different scenarios were considered based on the different combination of features such as word, part of speech tags of tokens. Since feature selection plays a vital role in CRF and SVM framework, experiments were carried out to find out most suitable features for AO tagging task. In the first scenario, all features which all features which include two words from left and two words from right of the current word without POS tags were used. In the second scenario, we used all features which include two words from left and two words from right of the current word within corresponding POS tag of each tokens as a feature. In the third scenario, we used one word from left and oneword from right of the current word within corresponding POS tag of each tokens as a feature.

The better result of CRFs and SVM methods are 87.4% and 86% is achieved respectively. This experimental result shows that the performance of developed chunker is significantly good for first time of AO text.

5.2. Contribution of the work

- ✓ 1,000 sentences dataset within POS tagged and chunk category were prepared.
- ✓ Chunking model was investigated with CRFs and SVM for AOL.
- ✓ The CRFs approach result shows better accuracy for AOT chunkingthan Support vector machine approaches.
- ✓ Important features in AO Text chunking were identified in both CRFs and SVM.

5.3. Future works

The study has shown that AO chunking can be done automatically using conditional random field algorithm and Support vector machine learning algorithm.The below pointsareimmediate recommend:

- ➢ With regard to dataset, AOL didn't have dataset which is manually tagged with POS and chunk categories. For this study, small amount of corpus is tagged manually but this corpus is insufficient and very small. So, in the future huge amount of dataset for AO chunking has to be prepared and the system has to be trained on that to improve its performance.
- ➢ Improve this research with others techniques.

Reference

[1]. Peter Jackson and Isabelle Moulinier, *Natural language processing for online applications*: text retrieval, extraction and categorization, 1984

[2]. Daniel Jurafsky& James H. Martin. Speech and Language Processing: *An introduction to natural language processing,computational linguistics, and speech recognition*. 2006

[3]. Karen Sparck Jones,*Natural Language Processing, a historical review*, University of Cambrige, Cambrige, October 2003.

[4]. Census Report (2017) "Ethiopia's population now 92 million"

[5]. TilahunGamta, *The Oromo language and the latin alphabet*, Journal of Oromo Studies, 1992.http://www.africa.upenn.edu/Hornet/Afaan_Oromo_19777.html last visited on Friday, October 31, 2014.

[6]. Tabor Wami's, New book titled, YewugenaDirsetochenaYetarik Ewunetawoch,2004.

[7]. WorkinehTesemaa, "*Designing a Rule Based Disambiguator for Afan Oromo Words".* In: American Journal of Computer Science and Information Technology, 2017

[8]. Abeba Ibrahim. "*A Hybrid Approach to Amharic Base Phrase Chunking and parsing"* (Unpublished Master‟s thesis). Addis Ababa University, Addis Ababa, Ethiopia. 2013.

[9]. GaddisaH .*"Automatic syntactic parser for Afaan Oromo complex sentence using context free grammar"*Addis Ababa university, Unpublished, October 2016.

[10]. Diriba M, *"An Automatic Sentence Parser for Oromo Language Using Supervised Learning Technique"*, Addis Ababa university, Unpublished, June 2002.

[11]. S. P. Abney. *Parsing by chunks* (pp. 257-278). Springer Netherlands. 1992.

[12]. Daelemans, W., S. Buchholz, and J. Veenstra. 1999. *Memory-based shallow parsing. In proceedings of CoNLL*, Bergen, N. submitted.

[13]. Misganu T. and Ramesh B, "*A Novel Anaphora Resolution Model for Afaan Oromo Language Texts."* In: International Journal of Engineering Research & Technology (IJERT), 2019.

[14]. Nabil KHOUFI. Et al.., "*Chunking Arabic Texts Using Conditional Random Fields*"

[15]. Akshay Singh, S M Bendre, and Rajeev Sangal. *HMM based chunker for Hindi. In: 2009 Proceedings of the Second International Joint Conference on Natural Language Processing*, October 2005.

[16]. R. Koeling (2000). *Chunking with maximum entropy models. In Proceedings of the 2nd workshop on Learning language in logic* and the 4th conference on Computational natural language learning-Volume 7 (pp. 139-141). Association for Computational Linguistics.

[17]. BiplavSarma,Anup Kumar*,"A Comprehensive Survey of Noun Phrase Chunking in Natural Languages"*International Journal of Engineering Research & Technology (IJERT) Vol. 4 Issue 04, April-2015.

[18]. Guang-Lu S. Et al.., "*Chinese Chunking Based on Naive Bayes Model"*. In: Research Institute of Information Technology, Tsinghua University, Beijing, China, 2010.

[19]. Asif Ekbal and Sivaji Bandyopadhyay,2010, *Named Entity Recognition using Support Vector Machine*: A Language Independent Approach: International Journal of Electrical, Computer, and Systems Engineering 4:2 2010.

[20]. Joel Mickelin,2013, *Named Entity Recognition with Support Vector Machines*: MSc Thesis, Royal Institute of Technology School of Computer Science and Communication, Stockholm, Sweden.

[21]. Chih-Wei Hsu and Chih-Jen Lin, *"A Comparison of Methods for Multi-class SupportVector Machines"* 2012

[22]. Roman Klinger, et.al. *Classical ProbabilisticModels and Conditional Random Fields. Algorithm Engineering Report*, TR07-2-013, ISSN 1864-4503. Dortmund University of Technology, Department of Computer Science, Germany, 2006

[23]. J. Lafferty, A. McCallum, F. Pereira, *"Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,"* Proceedings of the Eighteenth International Conference on Machine Learning (lCML).

[24]. Kuanghua Chen., Hsin-Hsi Chen. (1993). *A probabilistic chunker*. In: Proceedings of ROCLING-93, p. 99–117.

[25]. Chakraborty, Neelotpal., Malakar, Samir., Sarkar, Ram., Nasipuri, Mita. (2016). *A rule-based approach for noun phrase extraction from English text document*. In: Seventh International Conference on CNC-2016, p. 13–26.

[26]. Guang-Lu S. Et al.., *"Chinese Chunking Based on Maximum Entropy Markov Models*". In: The Association for Computational Linguistics and Chinese Language Processing on June 2006 pp. 115-136.

Appendices

Appendix 1: POS tagged used in this study

No	Tag	Description	Example
1	**NN**	**Common noun**	**Nama, Muka, jimmaa**
2	**NNP**	**A tag for all types of plural nouns that are not joined with other categories in sentence (not show location and direction)**	**Namoota,biyyoota, Namni,sangichi,**
3	**NNPS**	**Common noun plus pre/postposition (which is show location and direction)**	**InniBooranattii gale.**
4	**PN**	**proper nouns**	**Dammee,Caaltuu, Oromiyaa**
5	**PNS**	**Proper nouns that are joined with other postposition in sentence (not show location and direction)**	**Tolaan, caaltuun**
6	**PPN**	**Personal pronoun**	**Ana/ani, nuyi/nuti, sii/ati, isin, ishee**
7	**POP**	**Possessive Pronoun**	**Koo, kee, keenyaa**
8	**VB**	**Verbs**	**Kottu, Deemi, dubbisi, qaba**
9	**JJ**	**Adjectives**	**Bareedduu,diimaa, magaala,**
10	**AD**	**Adverb**	**Kaleessa,edana, yoomiyyuu,sirritti**
11	**AP C**	**Adpositions, pre-/postposition and conjunctions**	**Ni, irraa, garuu, fi, akka,yoom,ykn, bira, gara,**
12	**DT**	**Determiners**	**Kun, kana, kanneen, sana**
13	**CN**	**Cardinal numbers.**	**Lama, kudhan, 2012**
14	**PUNC**	**Punctuations**	**. ? ! ' : [{ ( ) =-__**

15	**NEG**	**Negative word**	**Hin,miti**
16	**PPNS**	**Personal pronoun + postposition**	**Nurraa**
17	**ADPP**	**Adverb plus postposition "-ttii"**	**Tulluunqawweedhukaasee [*fagootti*] bineensaajjeese**

Appendix 2: chunk tag used in this study

No	Chunk tag	Description
1	B-NP	Begin of noun phrase
2	I-NP	Inside of noun phrase
3	B-VP	Begin of verb phrase
4	I-VP	Inside of verb phrase
5	B-AdjP	Begin of adjective phrase
6	I-AdjP	Inside of adjective phrase
7	B-PP	Begin of postposition phrase
8	I-PP	Inside of postposition phrase
9	B-AdvP	Begin of adverb phrase
10	I-AdvP	Inside of adverb phrase
11	O	Outside of adverb phrase

Appendix 3: Sample chunk tag text for the training data set

Inni	PPN	B-NP
sa'a	NN	B-NP
Aannanii	NNP	I-NP
Bitate	VB	O
.	PUNC	O
Dallaan	NNP	B-NP
Loonii	NNP	I-NP
Kaleessa	AD	B-VP
Ijaarame	VB	I-VP
.	PUNC	O
Tolaan	PNS	B-NP
Nama	NN	B-NP
Ambooti	NPP	I-NP
.	PUNC	O
Gammachuun	PNS	B-NP
Suuta	AD	B-VP
Deema	VB	I-VP
Darbe	VB	I-VP
.	PUNC	O
Caaltuun	PNS	B-NP
Kaleessa	AD	B-VP
Daftee	AD	I-VP
Deebite	VB	I-VP
.		O
Bariisoon	PNS	B-NP
Garmalee	AD	B-VP
Soorame	VB	I-VP
.	PUNC	O

Appendix 4: Feature extracted:
Listed sentences:

```
[['Namooni', 'NNP', 'B-NP'],
 ['guguddaa', 'JJ', 'B-AdjP'],
 ['saddeeti', 'CN', 'I-AdjP'],
 ['dhufan', 'VB', 'O'],
 ['.', 'PUNC', 'O']]
```

Feature extracted file of the above listed sentences in all features set in CRFs:

```
[{'bias': 1.0,
 'current_token.lower()': 'namooni',
 'current_token.isdigit()': False,
 'current_postag': 'NNP',
 'First_token': True,
 'next_1_token_lower': 'guguddaa',
 'next_2_token_lower': 'saddeeti',
 'next_1_pos': 'JJ',
 'next_2_pos': 'CN'},
{'bias': 1.0,
 'current_token.lower()': 'guguddaa',
 'current_token.isdigit()': False,
 'current_postag': 'JJ',
 'prew_1_token_lower': 'namooni',
 'prew_1_pos': 'NNP',
 'next_1_token_lower': 'saddeeti',
 'next_2_token_lower': 'dhufan',
 'next_1_pos': 'CN',
 'next_2_pos': 'VB'},
{'bias': 1.0,
 'current_token.lower()': 'saddeeti',
 'current_token.isdigit()': False,
 'current_postag': 'CN',
 'First_token': True,
 'prew_2_token_lower': 'namooni',
 'prew_1_token_lower': 'guguddaa',
 'prew_2_pos': 'NNP',
 'prew_1_pos': 'JJ',
 'next_1_token_lower': 'dhufan',
 'next_2_token_lower': '.',
 'next_1_pos': 'VB',
 'next_2_pos': 'PUNC'},
{'bias': 1.0,
 'current_token.lower()': 'dhufan',
 'current_token.isdigit()': False,
 'current_postag': 'VB',
 'First_token': True,
 'prew_2_token_lower': 'guguddaa',
 'prew_1_token_lower': 'saddeeti',
 'prew_2_pos': 'JJ',
 'prew_1_pos': 'CN',
 'next_1_token_lower': '.',
 'next_1_pos': 'PUNC'},
{'bias': 1.0,
 'current_token.lower()': '.',
 'current_token.isdigit()': False,
 'current_postag': 'PUNC',
 'First_token': True,
 'prew_2_token_lower': 'saddeeti',
 'prew_1_token_lower': 'dhufan',
 'prew_2_pos': 'CN',
 'prew_1_pos': 'VB',
 'last_token': True}]
```

Apendix 5: Sample predicted chunk tagged output

Input file:

```
[('Tolaan', 'PNS'), ('gara', 'APC'), ('manaa', 'NN'), ('deeme', 'VB'), ('rafe', 'VB'), ('.', 'PUNC')]
```

Predicted output:

```
[(('Tolaan', 'PNS'), 'B-NP'),
 (('gara', 'APC'), 'B-PP'),
 (('manaa', 'NN'), 'I-PP'),
 (('deeme', 'VB'), 'B-VP'),
 (('rafe', 'VB'), '
```

NegessaWayessaJuka, et. al. "Ethiopian's Prominent LanguageAfaan Oromo Text Chunking Using Conditional Random Fields and Support Vector machines." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 22(3), 2020, pp. 44-61.