DEBRE BIRHAN UNIVERSITY

COLLEGE OF COMPUTING

DEPARTMENT OF INFORMATION SYSTEMS

Job Vacancy Announcement Text Categorization using Machine Learning Algorithm

PREPARED BY:

ALEMAYEHU TEFERA

May, 2021

DEBRE BERHAN, ETHIOPIA

DEBRE BIRHAN UNIVERSITY

COLLEGE OF COMPUTING

DEPARTMENT OF INFORMATION SYSTEMS

JOB VACANCY ANNOUNCEMENT TEXT CATEGORIZATION USING MACHINE LEARNING ALGORITHMS

A THESIS SUBMITTED TO THE COLLEGE OF GRADUATE STUDIES OF DEBRE BERHAN UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SYSTEMS.

BY

ALEMAYEHU TEFERA

May, 2021

DEBRE BERHAN, ETHIOPIA

DEBRE BIRHAN UNIVERSITY

COLLEGE OF COMPUTING

DEPARTMENT OF INFORMATION SYSTEMS

JOB VACANCY ANNOUNCEMENT TEXT CATEGORIZATION USING MACHINE LEARNING ALGORITHMS

BY

ALEMAYEHU TEFERA

NAME AND SIGNATURE OF MEMBERS OF THE EXAMINING BOARD

TITLE NAME SIGNATURE DATE

1. ADVISOR DR. MILLION MESHESHA , ______________, ________

2. CHAIRPERSON __________________________, ______________, ________

3. EXTERNAL EXAMINER __________________________, ________, ________

4. INTERNAL EXAMINER __________________________, __________, _________

DECLARATION

I declare that this thesis, which I submit to the department for examination is my original work and has not been previously presented in this or any other institution for a degree, diploma or other qualifications. All the material sources used in this work are duly acknowledged.

ACKNOWLEDGEMENT

First of all I would like to thank the almighty GOD with his mother virgin Mariam for everything. My special gratitude goes my advisor Dr.Million Meshesha for his constructive comments, suggestions and full devotions from beginning to end of my thesis work.

I am also grateful to my parents' dear dad and mom, you are the reasons and bases for me to be here today.

My deepest gratitude goes to my soul mate, without your encouragement, support and contribution this work couldn't come into being. I would like to say thank you in advance dear Ashu, your value is big in this work.

Lastly thank you all my classmate, friends, families who has been involved directly or indirectly on my work by advice, support or encouragement.

Alemayehu Tefera

May, 2021

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ANN	Artificial Neural Networks
DT	Decision Tree
DF	Document Frequency
GMM	Gaussian Mixture Model
HTML	Hyper Text Markup Language
IDF	Inverse Document Frequency
IG	Information Gain
KNN	K-Nearest Neighbor
LSI	Latent semantic indexing
ML	Machine Learning
NLP	Natural Language Processing
NB	Naïve Bayes
PCA	Principal Component Analysis
SVM	Support Vector Machine
SOFM	Self-organizing map
TF	Term Frequency
TFIDF	Term Frequency Inverse Document Frequency

ABSTRACT

Availability of large amount of electronic job vacancy text on the web makes the identification of relevant vacancy announcement related to a specific topic is a challenging task. It’s also true for Amharic texts. Amharic (አማርኛ) is an Ethiopian language which comes from Semitic language and used as first language by Amhara and working language of federal government. Large amount of electronic texts in this domain has been generated. So, a text categorization mechanism is required for finding, filtering and managing the rapid growth of online information. The goal of automatic text categorization is to classify documents into a certain number of predefined categories by using rule based or machine learning. The aim of this study is therefore to investigate the application of machine learning techniques for vacancy text categorization.

A total of 1678 vacancy announcement text with eight categories: “ጤና” (health), “ምህንድስና” (engineering), “የኮምፒዉተር ሳይንስ ዘርፎች” (computing), “ተፈጥሮ ሳይንስ” (natural science) “ማህበራዊ ሳይንስ” (social science), “ህግ” (law), “ግብርና” (agriculture) and “ቢዝነስ እና ኢኮኖሚክስ” (business and economics) were collected. After preprocessing the text for tokenization, stemming word variants and removing stop words and unwanted characters and weighting the importance of a term, 1610 pre-categorized text were used to train the classifier. In this study three supervised machine learning classifiers, namely support vector machine, k Nearest Neighbor and Naïve Bayes classifiers are used to categorize the vacancy text.

Experimental result shows that, Support Vector Machine outperforms the other two classifiers (K-Nearest Neighbor and Naïve Bayes) with an accuracy of 76.4%. This is a promising result to design vacancy text categorization model for jobs announced in Amharic language. ህግ (law) category is an item which performs the best classification accuracy in the current study. Because, law category is an item that share the least common terms with other field of study when compared with the rest of an items used in the current study. However, there are challenges in designing job vacancy text categorization model. The main challenge in this study is; there are conflicting tags as a result of common words in different categories where it is challenging task for machine to categorize these words. It is therefore recommended to apply semantic based Amharic vacancy text categorization.

Keywords: Job *Vacancy; Amharic language; Text Categorization****;*** *Machine Learning*

CHAPTER ONE

INTRODUCTION

1.1 Background

Web contains multimedia contents like text, audio, video, and graphics. As there is a great increase in the production of these electronic web contents, users find it difficult to obtain useful information from these contents. In the last few years, it has been seen that an exponential growth in the volume of text documents available on the Web in different domains [1]. These web documents contain rich textual information; more specifically, manuscripts, jobs, newspapers, journals, magazines, thesis and dissertations are available in different formats such as text, audio, video, and graphics [1]. But they are so numerous that users find it difficult to obtain useful information from them [2].

However, searching information of one's need in this large collection of documents requires organization and indexing [1]. Supporting the target users to access and organize the enormous and widespread amount of information is becoming a primary issue [3]; so that the need to categorize information resources has becoming an important issue [4] [2].

Text classification and categorization has become an active research area in information science that develops methods for assigning text documents to a pre-defined set of categories [2]. Categorization is the process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data, objects whose class label is known [5].

Currently, out of the abundant text documents available in electronic form on the web. Vacancy is one of the area in which very large documents are generated manually in newspaper as well as on web in an electronic form. As noted by Surafel [1], the web contains over a billion documents which is difficult to organize and analyze as well as Millions of people send e-mail every day. These collections and many other documents generated constantly on the web represent a massive amount of information [1].

Many techniques have been applied for text categorization. Most of the time text categorization process is done manually. In a manual text categorization (and classification) organizations define subjective categories, based on certain preferences and assign documents they write or receive to these categories according to this subjective definition of categories or classes. Manual categorization therefore, is based on human judgments. Organizations and users search for information and save it in categories meaningful to themselves. Although, the manual approach is accurate, it is time consuming and inconsistent. With fast growth in online document data, this would become more difficult with time [4].

Since there are a large collection of electronic documents generated today, it is difficult to manage and classify these documents by the traditional classification method. Therefore, automatic text classification and categorization of the documents in the domain is critical issue. Automatic classification techniques use algorithms that learn from human classification techniques. Its goal is the classification of documents into a fixed number of predefined categories [4].

Automatic text classification assigns the text to its pre-defined categories according to the main idea or subject. Today, automatic text classification is necessitated due to very large amount of text documents that we have to deal with daily [6]. It involves the machine learning methods which currently have been mainly used to develop text categorization model [7].

Machine Learning (ML) is the ideal solution in the case where a sufficiently large set of previously classified texts is already available — a so-called “training corpus”: the corpus is supplied to the ML algorithm, which learns autonomously what are the best strategies for classifying documents [8]. The most commonly used machine learning methods for classifying textual information includes supervised learning and unsupervised learning. Supervised machine learning methods are applied to develop a model that divides and categorizes a text into its categories [9] [10] [11]. The constructed automatic text categorization model then helps to decide and label topical labels to content to solve the problem of overloaded information.

Text Clustering is a text mining technique which divides the given set of text documents into significant clusters. It is used for organizing a huge number of text documents into a well -organized form. In clustering techniques, more similar clusters are grouped together than in other clusters. It improves efficiency and effectiveness of text categorization system which resulted in saving space, time and increase quality [12] [7]. It works with unlabeled texts those are easily available in the world.

Supervised machine learning algorithm uses pre-labeled examples as a training data. A machine learning algorithm can then learn the different associations between pieces of text and that a particular output (i.e. tags) is expected for a particular input (i.e. text) [13]. The supervised learning model is applied to automatically decide categories of data whose category is unknown [7]. Several techniques have been used for text classification such as Decision Tree (DT), Naïve Bayes (NB), K-Nearest-Neighbor (k-NN) and Support Vector Machine (SVM) [14].

In general, text classification includes topic based text classification and text genre-based classification. Topic-based text categorization classifies documents according to their topics [7]. Texts can also be written in many genres, for instance: scientific articles, news reports, movie reviews, and advertisements. Genre is defined on the way a text was created, the way it was edited, the register of language it uses, and the kind of audience to whom it is addressed. Previous work on genre classification recognized that this task differs from topic-based categorization [7].

There are two main categories of text classification; flat text classification and hierarchical text classification [15]. In flat text classification, there is no linkage that defines the relationship of each category as each category is processed separately.

Single classifier is trained to categorize a new document to certain classes. On the other hand, hierarchical text classification is used to classify large text documents by using divide-and-conquer approach to overcome a problem of large classification [16]. It decomposes the classification task into a set of simpler problems, one at each node in the classification tree that leads to more accurate classifier. Document classification to their predefined categories requires a large amount of hand labeled texts which is difficult and time consuming.

Therefore, the aim of this study is to develop a model for vacancy text categorization using supervised machine learning algorithms. The result of this research will have a significant role for job seekers by making their searching experience easier to find jobs categorically and for organizations for further data analysis.

1.2 Statement of the Problem

In the current day, large amount of vacancy announcement has been generated on the web. This increases the amount of texts that are available on the web in electronic form which is difficult to organize and manage. In the existing system the concerned body prepared the content of the job; i.e. job ID, title of the job, publishing date, required experience, educational background, job descriptions and related information. The job is then categorized based on its ID, title, experience (and/or based on the rule) that the expert defined and saves it accordingly, so that it can be retrieved later by its ID, title, date, experience and category when needed. Since manual categorization is based on human judgments; it is accurate. But it is time consuming and inconsistent [17].

In order to have easy access for ups-to-date and timely information, vacancy announcement should be organized in systematic manner. The greater our ability to store information, the more attention must be paid to the problem of organizing and retrieving it. In the last few years, automatic text classification systems have proven to be just as accurate, correctly categorizing over 90% of the text documents. They are also far faster and more consistent, so there has been a switch from manual to automated systems [1].

A job title is an all-encompassing very short form description that conveys all of relevant information relating to a job [1]. So based on the title the announcement can be categorized in to its predefined classes. But job title does not give full information for classification [18].

In this study our focus is on text categorization of jobs to a predefined set of occupation categories. Machine learning techniques have been successfully applied to text classification and categorization. The learning approach to document classification entails assigning documents (which may be images, text, or other entities) to a label of a predefined class or category to create a set of training data. This training dataset is used to learn a model that is then utilized to assign a class to new documents [14].

Many local researchers applied text classification in the news domain [3] [4] [19]. However, there is no attempt made by local researchers for categorizing job vacancy announcements using local languages, like Amharic. On the contrary, there are works done by foreign scholars. Lynch [20] tried to construct a model to predict job titles based on job descriptions using Random Forest and Support Vector Machines classification algorithms. In addition, Flora et al. [18] attempted to classify web job advertisements against a standard classification system of occupations by taking job titles as input for the classification process, but job title does not give full information for classification. There are other descriptors of job vacancy announcement for effective categorization. Such job vacancy descriptors include job qualification and educational background.

It is therefore the intension of this study to categorize vacancy announcements based on job qualification or educational background as an input using supervised machine learning techniques.

To this end, this study attempts to explore and answer the following research questions.

- Which supervised learning algorithm is suitable for categorizing job vacancy announcements?
- To what extent the proposed model is able to predict category of job

vacancy?

1.3 Objective of the study

1.3.1 General objective

The general objective of this study is to construct a model for the categorization of Amharic job vacancy announcement text using machine learning algorithm.

1.3.2 Specific objectives

To achieve the general objective of the study, the following specific objectives are formulated.

- ❖ To review related works in text classification to have a conceptual understanding of methods and algorithms used for text classification.
- ❖ To collect data from the web and apply preprocessing tasks in order prepare dataset.
- ❖ To select suitable classification algorithms for experimentation.
- ❖ To construct a classification model for Amharic job vacancy texts categorization.
- ❖ To evaluate the performance of the proposed model using effectiveness measures.
- ❖ To report finding of the study and recommend for the upcoming research area.

1.4 Significance of the study

This study will primarily support job seekers by making their searching experience easier to find jobs categorically concerning their profession. In addition, it helps job seekers, who visit job posting websites to check for active vacancies to find the jobs they need easily. Furthermore, the result of the research enables data to be aggregated, which in turn enables analysis for the organization. This gives valuable insights to employers as they design their recruitment strategies. Also the output of this research can be used as an input to the development of full-fledged automatic jobs classification for organizations (job posting websites). The current study also motivates other researchers to propose on text classification with the same domain using different approaches for applying the complete systems towards text categorization.

1.5 Scope and limitation of the study

The aim of this study is to categorize vacancy text by using supervised machine learning techniques. The scope of this study is limited to design text categorization model for Amharic vacancy text which mainly help job seekers when searching for jobs by simplifying their searching experience. In addition, this study is limited to include vacancies that require some qualification (educational background) which include job positions like gardener, cleaner. Vacancy text can be categorized by different parameters like position or job title, qualification of educational background, address, job type, experience, the focus of this study is to classify the vacancy text using qualification or educational background of Amharic language vacancy announcements. Some of the educational background like “economics” used direct English word in Amharic when announcing the vacancy. As a result, the Amharic stemming algorithm is not working for such words and needs modification which is left for further research.

1.6 Methodology

Methodology provides an understanding of how the proposed research is conducted in order to obtain information for developing the proposed systems [3]. It shows the path through which the researchers formulate their problem and objective and present their result from the data obtained during the study period. It contains tools and techniques that can be used for conducting the study. In the current study the following methodology is followed.

1.6.1 Research design

In the current study an experimental research is used as a general approach. An experimental research is a study in which the investigators formulate the experimental setup like how many experiments needs to be conducted, with what algorithms, parameters, weights and dataset [21]. In an experimental research repetitive experiments are used as analytical method. The validity and reliability of the study in this research type can be checked through testing and evaluations. In the current study the researcher reviewed all necessary journal articles, thesis report, conference papers, books and the Internet thoroughly for achieving the research objective. To conduct an extensive experiment, tasks such as data preparation, implementation and evaluation are performed as described below.

1.6.2 Data Preparation

The dataset used for conducting the experiment in this study is collected from job vacancy announcement websites and a total of one thousand six hundred seventy eight (1678) vacancy texts were extracted from these websites as a dataset in the current study. Important preprocessing tasks such as text cleaning, tokenization, normalization, stop word removal, stemming, and term weighting are performed using NLTK (Natural Language Toolkit) in Python to clean the data and prepare training and testing data set for running machine learning algorithms.

1.6.3 Implementation tools

In the current study, python programming language is used for classifying vacancy text. Python is one of the easiest languages to learn and use, while at the same time being very powerful: It is one of the most used languages by highly productive professional programmers and also Python is a free programming language [22]. It’s syntax is clear and readable where both experts and beginners can easily understand the code. Because the block structures in Python are defined by indentations, it has much less likely to have bugs in codes caused by incorrect indentation. It is also simple to get support and fast to code. Python provides fast feedback in several ways. Python programming encourages program reusability by implementing modules and packages. A large set of modules has already been developed and is provided as the standard python library, which is part of the Python distribution. Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction [23].

1.6.4 Evaluation methods

In this study the performance of the model is measured by using four classification evaluation metrics (accuracy, precision, recall and f-measure). The accuracy metric measures the ratio of correct predictions over the total number of instances evaluated. Precision metric is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class. Recall is used to measure the fraction of positive patterns that are correctly classified. A high recall means that the majority of the ‘positive’ items were labeled as belonging to the class ‘positive’. The f-measure (or f-score) combines the precision and recall to give a single score, and is defined to be the harmonic mean of the precision and recall [24]. These evaluation methods are selected because they are the most widely used ones to measure effectiveness of classification model created using machine learning algorithms.

1.7 Organization of the Thesis

This thesis report organized into five chapters. Chapter one introduced the overview of the study, a statement of the problems, objective, methodology, scope and limitation, and significance of the study is discussed. In the second chapter an overview and definition of text categorization, basic concepts in automatic classification, approaches to text categorization, text categorization phases, application of text categorization and related work is discussed clearly. In the third chapter an overview of Amharic language and its writing system and problems in Amharic writing system is discussed in detail. In the fourth chapter the general architecture of the proposed model for vacancy text categorization, data collection method, data preparation tasks, machine learning algorithms selected in the current study and performance evaluation method is discussed. In fifth chapter the experimental result of the proposed models with selected algorithms, discussion of the result, and comparison of the algorithm results and finally in the sixth chapter conclusion of the study result and recommendations are discussed.

CHAPTER TWO
LITERATURE REVIEW

2.1. Overview

Nowadays, with the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data. In order to address the problems of manual categorization, automatic ways are explored as an alternative approach using rule-based or machine learning (ML) techniques. The goal of automatic text categorization is to classify documents into a certain number of predefined categories [25]. Classification algorithms of machine learning facilitate the process of categorization. Compared to manual classification, machine leaning approach offers the advantages of automation, efficiency, and consistency [4] [5].

Given C= {c1, c2…cn} is a set of categories (classes) and D= {d1, d2,…,dm} is a set of documents, text categorization is the task of assigning ci to dj (1≤i≤n and 1≤j≤m) a value of 0 if the document dj does not belong to ci; otherwise a value of 1. The mapping is also known as decision matrix [4]. In this chapter the concept of text categorization, approaches and application of text categorization as well as related works are discussed in detail.

2.2. Types of automatic text categorization

Categorization can be classified into different groups depending on the way categorization is done. There are single-label vs. multi-label categorization, document-pivoted vs. category-pivoted categorization and soft vs. hard categorization [5].

2.2.1. Single-label versus multi-label text categorization

For the set of classes **C, set** of documents **D** and for a given integer **k** where C= {c1, c2…cn} and **D** = {d_1, d_2, d_3, d_4, d_5, d_6, . . . d_m}, it is possible to consider the following, exactly **k** elements of **C** are assigned to each **d_j** Є **D**. Single- labeled (also called non overlapping categories) is the case when exactly one category (**k** = 1) must be assigned to each $\mathbf{d}_j$ Є D .and the case when any number of categories from **0** to **m** may be assigned to the same $\mathbf{d}_j$ Є **D** is called multi label (also called Overlapping categories) [1] [4] [5].

In the case of single-label text categorization only one predefined category to each "unseen" natural language text document is assigned and often defined as non-overlapping categories whereas multi label is when as many as possible category is assigned to the "unseen" natural language text document. And that is why it is called overlapping categories. Binary Text categorization is special case of single-label TC in which each dj Є D must be assigned either to category c_i or to its complement ci (complement sign).

2.2.2. Category-pivoted versus document-pivoted text categorization

There are two ways of using text classifier. Given a document $d_j \in D$, we might want to find all the categories $c_i \in C$, under which this document should be filed (document-pivoted categorization – DPC); on the other hand, given a category $c_i \in C$, we might want to find all the documents $d_j \in D$ that should be filed under this category (category pivoted categorization – CPC).

In document pivoted text categorization, it is given that $d_j \in D$ it have to find all $c_j \in C$ under which it should be filed. That means document d_j is searched under all categories and the required corresponding category will be found which contain given document d_j.

It is given that $c_i \in C$, we have to find all dj ∈ D under which it should filed. This means that a category c_i is searched under all documents and the required corresponding document will be found which contain given category $c_{i.}$

Document pivoted categorization is thus suitable if documents become available at different moments in time. For example filtering e-mail. On the other hand category pivoted categorization is suitable when; a new category c|C|+1 may be added to an existing set C={c1,…, c|C|} after a number of documents have already been classified under C , and these documents need to be reconsidered for classification under c|C|+1.

Document pivoted category is used more often than category pivoted categorization, as the former situation is more common than the latter [26].

2.2.3. “Hard” categorization versus “Soft” categorization

The hard categorization completely automates the text categorization in which it needs a true or false decision for each pair (dj, ci). It is a kind of decisions that are taken by autonomous text classifiers, or software systems that need to decide and act accordingly without human supervision [5]. Soft categorization, on the other hand, uses partial automation of the text categorization system which requires different methods [3]. In hard categorization the algorithm decides a value for each document-category pair $(d_j, c_i) \in D \times C$. A complete automation of the TC task requires a decision for each pair $\{d_j, c_i\}$. Ranking is when the algorithm ranks all categories in C according to how well the document fit into each category, a partial automation [1].

Given $d_j \in D$, a system might simply rank the categories in $C = \{c_1, c_2, c_3, c_4 \ldots, c|C|\}$. Assume that c_3 has the higher rank in terms of estimated appropriateness to d_j as compared to c1, c2, c4. Then without taking any hard decision on any category the final ranked list C_R will be:-

$C_R = \{c_3, c_1, c_2, c_4 \ldots, c|C|\}$

The great advantage obtained from ranked list would be it help to a human expert in charge of taking the final categorization decision, since it could thus restrict the choice to the category (or categories) at the top of the list, rather than having to examine the entire set. Alternatively, given ci $\in$ C a system might simply rank the documents in D according to their estimated appropriateness to ci; symmetrically, for classification under ci a human expert would just examine the top-ranked documents instead of the entire document set. These two modalities are sometimes called category-ranking TC and document ranking TC, respectively, and are the obvious counterparts of DPC and CPC [1] [27].

In critical applications when the effectiveness of a fully automated system may be expected to be significantly lower than that of a human expert; semi-automated interactive classification systems are useful in addition to these categorizations. This may be the case when the quality of the training data is low, or when the training documents of fully automated classifier cannot be trusted to be a representative sample of the unseen documents that are to come, which deteriorates the results and hence cannot be trusted [26].

According to Sebastiani [26], it is desirable to use hard categorization for automatic text categorization because the hard categorization fully automates the text documents of the specified language. The other approaches of text categorization are flat and hierarchical text categorization. In flat categorization, the single-label is the commonly used mechanisms of categorizing documents. In hierarchical text categorization, the multi-label text categorization mechanisms are used commonly. The document-pivoted and category-pivoted can be used based on the document occurrence of the specified time [3].

2.3. Approaches to text categorization

Text categorization can be performed in two approaches. These are manual text classification and automatic text classification where automatic text can be supervised, unsupervised and semi supervised learning [28] [10] [11].

2.3.1. Manual Classification

Manual text classifiers are built from labeled training (often manually) set of documents. Labeling is usually done manually by human experts (or the users) or human annotator interprets the content of text and categorizes it accordingly [29], which is a labor intensive and time consuming process. Domain experts who are thoroughly versed in the category structure or taxonomy are being used to label documents. Manual classification can achieve a high degree of accuracy-although even domain experts will occasionally disagree on how to categorize document. However, manual labeling of a large set of training documents is a bottleneck approach as it is a time consuming, more labor-intensive and therefore most cost than automated techniques [30].

2.3.2. Automatic Classification

Automatic Text Classification is the task of automatically assigning a given document to a set of pre-defined categories based on its textual content and extracted features. There are three approaches to automatic text classification, which can be rule based, machine learning based and hybrid approach [3] [29] [31].

Rule based techniques classify text document into organized groups by using a set of handcrafted linguistic rules. These rules instruct the system to use semantically relevant elements of a text to identify relevant categories based on its content. Each rule consists of an antecedent or pattern and a predicted category [29]. In rule based classification, keywords and Boolean expressions are used to categorize a document. This is typically used when a few words can adequately describe a category. For example, if a collection of medical papers is to be classified according to a disease, then a medical thesaurus that lists each disease together with its scientific, common and alternative names can be used to define the keywords for each category. While rule-based approach is effective for carefully tuning a limited number of categories, the expense of defining and maintaining categories is generally prohibitive for large-scale classification systems [4]. Although rule based approach is effective for carefully tuning a limited number of categories, the expense of defining and maintaining categories is generally prohibitive for large scale classification systems [4].

As discussed above rule based approach is relying on manually crafted rules. In other way text classification with machine learning learns to make classifications based on past observations. By using training data, a machine learning algorithm can learn the different associations between pieces of text and that a particular output (i.e. tags) is expected for a particular input (i.e. text) [29]. It is a data analytics technique that teaches computers to do what comes naturally to humans: that is learning from experience. Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. Machine learning algorithms adaptively improve their performance as the number of samples available for learning increases [32].

Hybrids systems combine a base classifier trained with machine learning and a rule-based system, which is used to further improve the results. These hybrid systems can be easily fine-tuned by adding specific rules for those conflicting tags that haven't been correctly modeled by the base classifier [29].

2.4. Machine Learning Approach

Machine learning approach can be supervised learning, unsupervised learning and semi-supervised learning [3] [5] [33]. Figure 2.1 depicts the major machine learning techniques and algorithms.

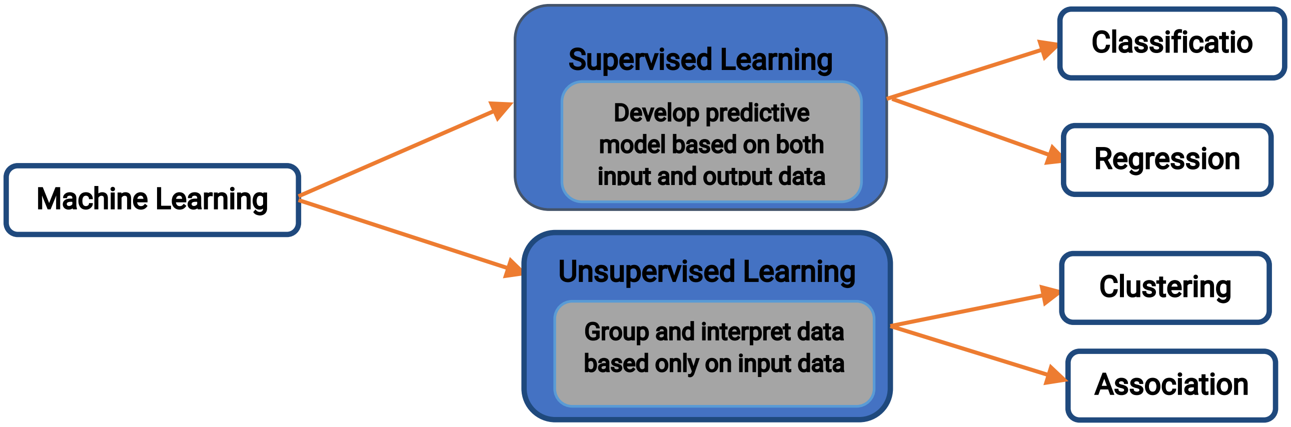

Machine Learning
Supervised Learning
Develop predictive model based on both input and output data
Unsupervised Learning
Group and interpret data based only on input data
Classificatio
Regression
Clustering
Association

Figure 2. 1 Machine learning algorithms [32] [34]

Supervised learning is one of the approaches in machine learning which uses a known dataset (usually called the training dataset) to make predictions. The training dataset includes input data (attribute values other than the class label) and response values (class label). From these input and output or response values, the supervised learning algorithm seeks to build a model that can make predictions of the response values for a new dataset. Using larger training datasets and optimizing model hyper parameters can often increase the model's predictive power and ensure that it can generalize well for new datasets. A test dataset is often used to validate the model [32].

The learning approach to document classification entails assigning documents (which may be images, text, or other entities) a label of a predefined class or category to create a set of training data. This training dataset is used to learn a model that is then utilized to assign a class to new documents [4] [14].

Unsupervised learning identifies a group, or clusters of related documents as well as the relationship between these documents. Commonly referred to as clustering, this approach eliminates the need for training sets because it does not require a pre-existing category structure. However, clustering algorithms are not always good at selecting categories that are intuitive to human users. For this reason, clustering generally works hand-in-hand with the supervised learning techniques [4].

Unsupervised learning finds hidden patterns or intrinsic structures in data. It is used to draw inferences from datasets consisting of input data without labeled responses. Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for cluster analysis include gene sequence analysis, market research, and object recognition [32]. With this learning approach, pre-classified documents are not required since the method tries to exploit regularities found in the document and make group or cluster based on similarity [19].

Most of the time, we need labeled data to perform supervised machine learning. Labeling a lot of data is expensive and time consuming. In another case unsupervised algorithms don't need labels but can learn from unlabeled data. Due to its time complexity and interpretation problems, using these unlabeled data for classification is not preferred. So there should be a mechanism needed to combine both clustering algorithms and classification algorithms using the process of unsupervised learning [35]

[36].

In ML terminology, the classification problem is an activity of supervised learning, since the learning process is “supervised” by the knowledge of the categories and of the training instances that belong to them [1].

Supervised learning uses classification and regression techniques to develop predictive models [32]. Classification techniques predict discrete responses. For example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition, and credit scoring. Classification techniques are used if the data can be tagged, categorized, or separated into specific groups or classes, whereas, regression techniques predict continuous responses. For example, changes in temperature or fluctuations in power demand. Regression techniques can be used for working with a data range or if the nature of the response is a real number, such as temperature or the time until failure for a piece of equipment.

In this study classification algorithms are applied for the purpose of vacancy announcement text categorization.

2.5. Text Categorization Phases

It is difficult to use text documents represented by natural language documents directly for building models using machine learning algorithms. In order to solve such a problem, natural language text document should be mapped into a schematic representation of its content. So that , the categorization algorithm changes each document into a vector of weights corresponding to an automatically chosen set of keywords [3] [33]. This transformation of documents has two main steps.

First, suitable representation of the document has to be chosen, which is used for all documents to be indexed and the representation has all necessary words that can characterize the documents.

Second, it assigns weights to each selected reprehensive terms which shows the frequency of occurrence of the term in the indexed document. Even though the document indexing is performed, still the results obtained has high dimension and take a large amount of storage space. So, techniques for dimensionality reduction should be applied. The main idea behind these techniques is to map each document into a lower dimensional space that can potentially take into account the dependencies between the terms. The following image illustrates the procedures followed for text classification.

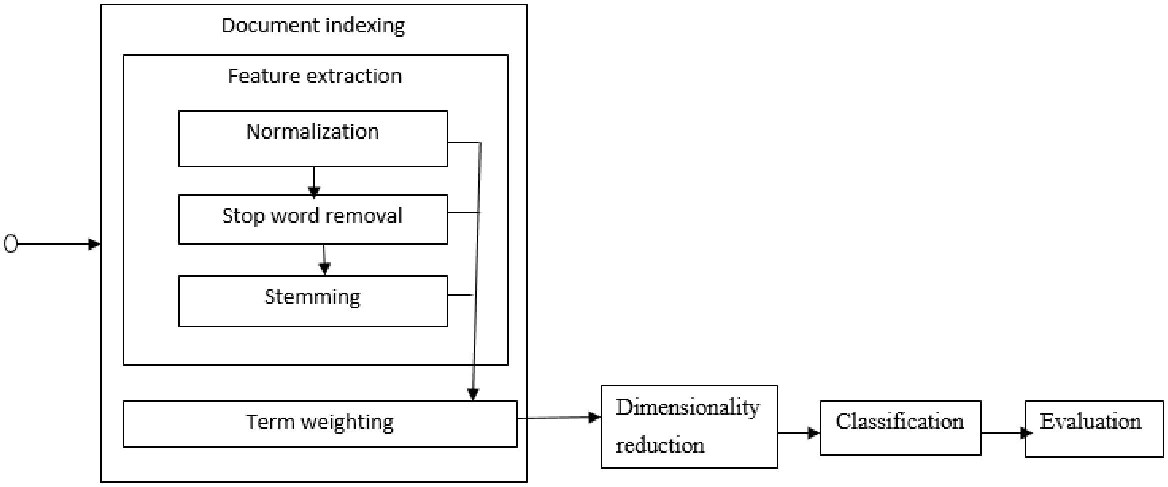


Figure 2.4 text categorization phases

- **Document Indexing**

The task of document indexing involves mapping of a document dj into suitable representation of its content that can be used for building the classifier algorithms. The choice of the representative term of a document depends on the individual choice of meaningful terms [15]. Generally, there are two main steps for indexing documents in the given corpus [26]. Representative terms are selected from the documents. After selecting the representative terms, the non-discriminating terms are removed from the documents in the given corpus. This process is also called feature selection step. As a result, most researchers use representative term and feature selection interchangeably [15]. The removed terms are both the frequently and very rarely appearing terms because such words or terms do not distinguish one document from other document in a given corpus.

Second, weight is assigned to each document which represents the documents by numeric vector. This vector includes the weight of the term in a document. So, the weight factor should represent the importance of the term for the categorization of the document. The most common term weighting approaches used in text categorization are Boolean weighting, term frequency weighting, and term frequency × inverse document frequency weighting [3]. There are certain concepts that need to be defined in any of term weighting approaches. Which are,

- tf_{ij} is the frequency of term i in document dj;
- Ni is the total number of documents in the document corpus;
- N is the number of documents in the corpus where term i appears; and
- |T| is the number of distinct terms in the document collection (after stop word removal and stemming is performed).

The simplest method of term weighting is Boolean weighting and it assigns 1(existence) if the term exists in a document or zero (absence) if the term does not appear in the document. Here, the weighting is only existence or absence that does not show in which documents the term appears more or less. For this reason it is not widely used approaches.

On the other hand, term frequency weighting approach counts the appearance of the term in documents. In this method, the weight of a term in a document represents or is equal to the number of times the term appears in the document. Sometimes, the most frequent term could not discriminate one document from other documents. If the term frequency of the term is high, its discriminating power to the mean documents is low. So, this term weighting techniques is no mostly used in text categorization processes. As a result, the term frequency × inverse document frequency weighting (Tf×idf) which uses the frequency of the most discriminating term in a given documents is most commonly used [15]. Here, the weight of term i in document d is proportional to the number of times the term appears in the document and inverse proportional to the number of documents in the corpus in which the term appears. Tf×idf function can be defined as follows:

$$Wtij=tfij* \log (N/Ni) \quad (2.3)$$

Where in the above equation 2.3, tf_{ij} is the term frequency of term i in a document j , log (N/Ni) is the inverse document frequency of the term, N is the total number documents in the corpus, and Ni is the number of documents term i appears.

The tf×idf weighting approach weights the frequency of a term in a given document with a factor that discounts its important if it exists in most of the documents. At the end of this process, the index file is constructed using indexing structure such as inverted file, signature file etc.

An index file stores a mapping from contents such as terms to its locations in a document or a set of documents. The purpose of indexing is to have fast searching mechanism when there are a lot of documents in the database. The most widely used indexing structure is inverted file which can be represented in two ways: an inverted index file containing a list of references to documents for each word and the inverted index file which contains the documents which the term appears, and the position of each word within a document [37].

- **Dimensionality reduction**

After the index file is generated the dimension of the index file is reduce in order to save the storage space and enhance the processing speeds. The dimensionality reduction maps each document into lower dimensional space. This improves the categorization performance of a given text documents. There are various dimensionality reduction techniques that can be classified as either supervised or unsupervised. Supervised dimensionality reduction techniques use the class-membership information for computing the lower dimensional space. Examples of supervised dimensionality reduction techniques are document frequency (DF) and information gain (IG). However, unsupervised dimensionality reduction techniques compute a lower dimensional space without using any class-membership information. These techniques are primarily used to improve the retrieval performance and rarely used for document categorization. Examples of such techniques include principal component analysis (PCA), latent semantic indexing (LSI), Kohonen self-organizing map (SOFM), and multidimensional scaling (MDS). However, the unsupervised dimensionality reduction methods are not commonly used in text categorization process [15]

- **Classifier learning**

A learning classifier is a function that maps an input attributes to the class membership it belongs to [15]. The attributes in this classifier are the list of terms found in the document, whereas the classes are the predefined categories to which the documents belong. In automatic text categorization, a text classifier for a given category is automatically generated by a learner. The learner observes the characteristic of a given document under a pre-defined category and determines the new unseen document to specified category. Most of the time evaluation procedure of learning classifiers in text categorization uses three data sets [15]. These are training set (Tr), validation set (Va), and test set (Te). The training set is the set of documents observed when the learner builds the classifier. After building the training set the validation set is used for choosing for a parameter on which the classifier depends and for evaluating the effectiveness. Finally, the test set is used to evaluate the effectiveness of the classifier. The test set is the set on which the effectiveness of the classifier is finally evaluated. Both the validation set and test set are used for evaluating the effectiveness of the classifier.

- **Evaluation**

Performance of the model is measured by using four classification evaluation metrics (accuracy, precision, recall and f-measure).

2.6. Classification algorithms

Naïve Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest

Neighbors (KNN), Neural Network (NN) are among the ML algorithms used for text categorization [4].

Naïve Bayes (NB)

This classifier uses the join probabilities of words co-occurring in the category training set and the document to be classified to calculate the probability that the document belongs to each category. It is a probabilistic classifier which uses the properties of Bayes theorem assuming the strong independence between the features [38]. The document is assigned to the most probable category (ies). Naïve Bayes classifiers have been proven in many domains, especially in text categorization, despite the simplicity of the model and restrictiveness of the independence assumptions it makes [5].

Naïve Bayes classifier has many advantages and it is easy to implement. Among the advantages of the Naive Bayes classifiers, it is simple technique results in high accuracy, especially when combined with other methods. Also this classifier requires small amount of training data to calculate the parameters for prediction. Instead of calculating the complete covariance matrix, only variance of the feature is computed because of independence of features. The limitations of this classifier is the independence and equally importance assumption which may cause skewed results, especially if many of the variables are interrelated, as that relation will have a greater effect on the decision, for better or for worse. Naïve Bayes classifier do not allow for categorical output attributes [39].

The Naive Bayes algorithm calculates the conditional probability P (C|D), where C is the class value and D is an instance of the sample. The D will be classified to the C if P (C|D) is the maximum one of all class values [39].

$$\mathbf{P}\,(C/D) = \frac{P(C/D)^{*}\,P(C)}{P(D)} \qquad \mathbf{(2.\,1)}$$

According to Bayes rule, P (D) is constant and assuming all the predictors are independent each other, the problem converts to calculate the maximum P(C/D) * P(C) which is equal to .$P(\prod_{k=1}^{n} p(D|C))P(C)$

In Naïve Bayes classifier text documents can be represented as document vectors using two models (i.e. the Multivariate Bernoulli Model and Multinomial model). In Multivariate Bernoulli Model the document vector is a binary vector which simply indicating the absence or presence of feature terms. A vector of binary attributes is used to represent document that indicating which words occur and do not occur in the document. The frequency of a word in a document is not captured. In Multinomial model document vectors additionally retain the information regarding frequency of occurrence of feature terms. A document is represented by the set of word occurrences from the document in which the number of occurrences of each word in the document is captured [5] [31] [17].

Decision Tree (DT)

Unlike NB classification, Decision Tree classification does not assume independence among its features. In a Decision Tree representation the relationship between attributes is stored as links. Decision tree (DT) can be used as a text classifier when there are relatively fewer number of attributes to consider, however it becomes difficult to manage for large number of attributes [31]. This classifier has an internal and terminal (leaf) nodes. The internal node indicates the different attributes of the text classification, and the leaf nodes show the classification of the attributes [33]. The internal nodes are labelled by the features, the edges leaving a node are labelled by tests on the feature's weight, and the leaves are labelled by categories. DT classifier categorizes document by starting at the tree root and moving successfully downward via the branches (whose conditions are satisfied by the document) until a leaf node is reached. The document is then classified in the category that labels the leaf node [40].

In decision tree-based feature ranking, a decision tree induction selects relevant features and ranks the features. Decision tree induction is decision tree classifiers learning, constructing a tree structure with internal nodes (non-leaf node) denoting an attribute test. The algorithm at each node chooses best attribute to partition data into individual classes. Information gain measure is used to choose the best partitioning attribute by attribute selection. Attribute with highest information gain splits the attribute [40]. Information gain is used to induction of decision tree using this formula [41]:-

$$\mathbf{Info(D) = -\textstyle\sum_{i=0}^{m} Pi*log2(Pi)} \qquad \mathbf{(2.\ 2)}$$

where pi is probability that an arbitrary vector in D belongs to class ci. A log function to base 2 is resorted to as information is encoded in bits. Info (D) is an average amount of information needed to identify the class label of tuple in D. The tuples D on some attribute A having v distinct value, {a1, a2, a3, av }, as observed from the training data. If A is discrete-valued, these values correspond directly to the v outcomes of a test on A. Attribute A can be used to split D into v partitions or subsets, {D1, D2, Dv }, where Dj contains those tuples in D that have outcome aj of A. Information gain is defined as the difference between the original information requirement and the new requirement. The attribute with the highest information gain, (Gain (A)), is chosen as the splitting attribute at node N.

Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning algorithm which can be used for both classifications and regression challenges. However, it is mostly used in classification problems [42]. SVM often considered as the classifier that produces the highest accuracy results in text classification problems. It creates a maximum margin hyper plane that lies in a transformed input space and splits the example classes, while maximizing the distance to the nearest cleanly split examples [4]. This shows that the value of the given parameter of hyper plane to the nearest training patterns from given classes is maximized as many training patterns as possible.

SVM is non-probabilistic binary linear classifier that uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (a decision boundary separating the instances of one class from another). Data from two classes can always be separated by a hyperplane, with an appropriate nonlinear mapping to a sufficiently high dimension. The rest of the training data have no influence on the trained classifier [43].

Although the training time for SVMs can be long (since it is computationally expensive and requires extensive memory space), they are highly accurate, owing to their ability to model complex nonlinear decision boundaries. Their use of support vectors for identifying decision boundaries makes them much less prone to overfitting than the other methods. Moreover, since they usually are subsets of the training, the support vectors provide a compact description of the learned model [4].

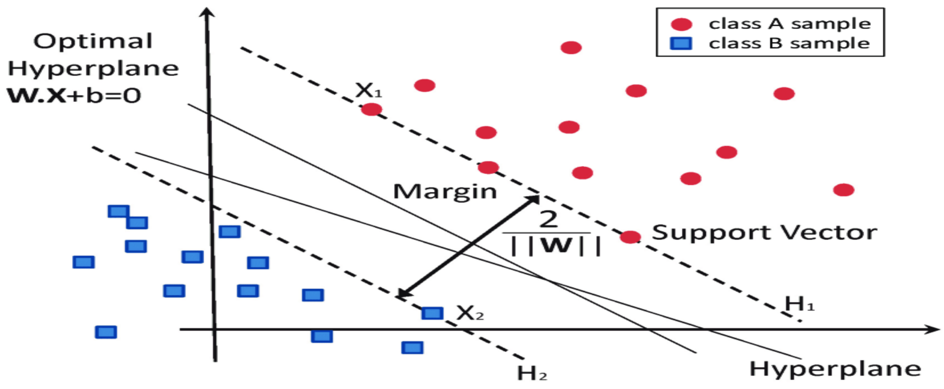


Figure 2. 2 Classification of data by Support Vector Machine [44]

As stated above the goal of this classifier is to find the best hyper-plane which separates the one class data points from the other class data points with the maximum possible margin for each set of points from the hyper-plane. The data points on the margins are called “support vectors”. SVM are well suited for problems with dense concepts and sparse instances. Most Text Categorization problems are linearly separable which, not being a limitation to SVMs, makes the computation much faster and simpler.

Support vector machine considers that each set of features represents a position inside a hyperspace then the SVM tries to divide it using a hyperplane maximizing the distance between this hyperplane and each vector, minimizing the objective function. This space division is hard to accomplish, and sometimes impossible, for this the SVM can use a margin that allows misclassifying some examples but increases the overall performance. The main advantages of SVM is its potential to handle large features and its robust when there is a sparse set of examples because most of the problem are linearly separable.

K-Nearest Neighbors (KNN)

K-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. It is non parametric method used for classification or regression. It is one of the simplest classification methods used in data mining and machine learning. It is the most accepted classification method due to its ease and practical efficiency. K-NN classifier doesn’t necessitate fitting a model and it has been proved to have superior performance for classifying several types of data [45] [46].

The rules of k-NN classification are created by the training samples alone with no other additional data. In a more complicated approach, k-NN classification, finds a group of k objects in the training set that are nearest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighborhood. The k-Nearest Neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space [KNN].

The Nearest Neighbor (NN) rule is the simplest form of KNN when K = 1. Given an unknown sample and a training set, all the distances between the unknown sample and all the samples in the training set can be computed. The distance with the smallest value corresponds to the sample in the training set closest to the unknown sample. Therefore, the unknown sample may be classified based on the classification of this nearest neighbor [45]. It works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label in the case of classification or averages the labels in the case of regression.

Neural Network (NN)

NN analysis is basically a prediction tool modeled on how human brain works [4]. It is an input output processing systems inspired from human brain that is they consist of neurons which is fundamental unit of the human brain. NNs are trained to recognize certain patterns or behavior when fed with a large data set and then they can determine predictors of a dependent variable. Thus, NN can be defined as a distributed processor that can create knowledge based on experience and make that knowledge available for future use [4]. Neural network is an assemblage of neurons with weightages which connects them, they process records one at a time and learn by comparing their classification with the actual classification. Neural network has the properties like robustness, self-learning and adaptiveness.

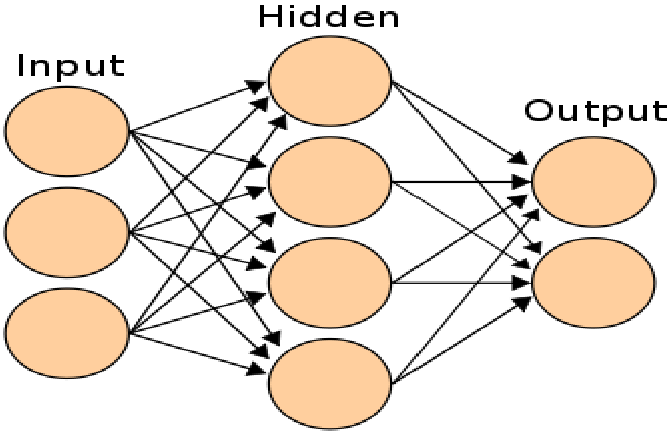


Figure 2. 3 Simple hidden layer neural network [47]

Neural network can be defined in three parts or layers they are input layer, hidden/ intermediate layer and output layer. The duty of the input layer is to receive the input signals from the outer system. Coming to the hidden layer it is comprised of neurons. The learning of the neural network is fully supervised hence for the input provides to the neural network has an answer or output [14]. The neural network takes input values and weights from the input layer as input and then it goes to the hidden layer in which a function sums the weights and maps the results to the corresponding output layer units. We can have 'n' number of hidden layers in between the input and output layers. Depending on the number of hidden layers the network will be named as single layer neural network or multi layered neural network (for more than one hidden layers) [48].The working of artificial neural networks is as follows [49] [50]

1. Information is fed into the input layer which transfers it to the hidden layer
2. The interconnections between the two layers assign weights to each input randomly
3. A bias added to every input after weights are multiplied with them individually
4. The weighted sum is transferred to the activation function
5. The activation function determines which nodes it should fire for feature extraction
6. The model applies an application function to the output layer to deliver the output
7. Weights are adjusted, and the output is back-propagated to minimize error

2.7. Applications of text categorization

Recent advances from IR and AI have made text categorization a hot research issue. Its use appears in a wide variety of applications [26].

Email filtering

Systems for filtering a person's incoming emails to weed out scam or spam or to categorize them into different classes are just now available.

Mail routing

Large enterprises are currently automating their document processing by means of workflow management system, allowing an image of the document to circulate through the company rather than the original. In particular, they aim for a uniform treatment of incoming mail, whether it is electronic or in paper form. A bottleneck in this approach is the entering of documents into the right work flow. This process involves a superficial interpretation of the contents of the document, which is time consuming and error prone.

News monitoring

In knowledge-based companies like the stock exchanges, numbers of people are concerned with the scanning of newspapers and other information sources for items which are concerned with the national or international economy, or with individual companies on the stock market. The results are sent to the person who should be informed.

Narrowcasting

Press agencies strive to give more and more individual service, where each client obtains out of the large stream of outgoing news items only those that are relevant to him, according to his profile.

Content classification

Large information brokers have traditionally used pre-classification of documents as an aid in document disclosure. Documents are manually given a place within a large semantically hierarchy, or index terms according to a given thesaurus. This process is costly and error prone and changes in the thesaurus are hard to accommodate. Modern search machines on the web use an automatic pre-classification of web pages.

2.8. Related works

There are many researches that have been conducted on text classification on different domains. Different approaches like manual text classification, rule based text classification and machine learning approach (which can be unsupervised and/or supervised) have been followed to conduct text categorization on each domain. They achieved the best accuracy on each of their works. News, email-filtering, news routing and medical document categorizations are the most common area in which text categorization is applied with different languages. In the next sub sections we will briefly review some of the text categorization works that have been conducted on different areas using different approaches.

2.8.1. Works done in text categorization

Jindal, Rajni, and Shweta Taneja [51] have proposed a novel lexical approach to text categorization in the bio-medical domain. They have proposed LKNN (Lexical KNN) algorithm, in which lexemes (tokens) are used to represent the medical documents. These tokens are used to classify the abstracts by matching them with the standard list of keywords specified as MESH (Medical Subject Headings). It automatically classifies journal articles of medical domain into specific categories. The collection of medical documents, called Ohsumed, were used as the test data for evaluating the proposed approach. Comparative study have been done to compare the performance of traditional KNN and LKNN with the different K values. And the result is noted in terms of f-measure for k=1,f-measure for KNN=1 and for LKNN=1;k=5, f-measure for KNN=0.75 and for LKNN=0.77;k=7, f-measure for KNN=0.76 and for LKNN=0.77;k=10,f-measure for KNN=0.78 and for LKNN=0.8;k=12,f-measure for KNN=0.8 and for LKNN=0.8;k=15,f-measure for KNN=0.83 and for LKNN=0.84. The results show that LKNN outperforms the traditional KNN algorithm in terms of standard F-measure.

Suleymanov [52] attempted to design automatic news labeling of Azerbaijani news articles corpus by applying supervised machine learning approach specifically, naive Bayes, support vector machines (SVM) and artificial neural networks classifiers. A total of 130000 news articles have been gathered along with their assigned categories. The documents are grouped under 8 mutually exclusive categories. Chi-squared test and LASSO (Least Absolute Shrinkage and Selection Operator) methods have been implemented for feature selection and pre-processing. By applying naive Bayes, the highest accuracy we got was 80.4%. This accuracy is achieved by applying count vectorization as feature extraction. The least accuracy observed was by using tf-idf approach with Naïve Bayes and artificial neural network model gave 86.3% accuracy result on Azerbaijani news article dataset. Application of feature selection namely, Chi squared test increased the accuracy of artificial neural networks by 2.8%.

Different local researches have been proposed on text classification using different approaches and domains. Most of these local works are done on news domain in different local languages like Amharic news, Afan Oromo news and Tigregna news.

Worku [19], has conducted research on Amharic news classification with the aim of classifying Amharic text news automatically using neural networks learning method called learning vector quantization. First text preprocessing techniques has been applied on the dataset which includes tokenization, removing of stop-words, and stemming. The remaining terms are organized according their frequency. Two weighting schemes, Term Frequency (TF) and Term Frequency by Inverse Document Frequency (TF*IDF), are used so as to weight the features in news documents to construct news by features matrix, which is fed to the learning algorithm and The result shows that the term frequency weighting scheme outperforms term frequency ×inverse document frequency weighting scheme by 3.54% on average .Using the TF weighting method, 94.81%, 61.61% and 70.08% accuracies are obtained with three, six and nine categories experiments respectively and the average accuracy is 75.5%. For similar experiments, Using TF*IDF weighting method an accuracy of 69.63%, 78.22% and 68.03% is obtained with three, six and nine categories experiments and an average accuracy is 71.96%.

Alemu [4], attempts a hierarchical text classification of Amharic news items using support vector machine. The research has an aim of constructing hierarchical classifier and it has evaluated the performance of the hierarchical classifier over the flat classifier with same dataset. The result of the experiment shows that the performance of the classifier increases as it moves down through the hierarchy. Besides, the hierarchical classifier out performs the flat classifiers with same data set.

Gebrehiwot [3], attempted to design a two-step Tigrigna text categorization system. First, clustering is techniques are applied to the data for finding natural grouping of the unlabeled Tigrigna text documents. Here, repeated bisection and direct k-means clustering algorithms are used to obtain documents of natural group of the Tigrigna data set. The repeated bisection clustering algorithm outperforms the direct k means clustering algorithms. So, the repeated bisection clustering algorithm results are selected for classification task. Second, decision tree and support vector machine techniques are used for classification task. The SMO support vector machine classifier performs better than J48 decision tree classifier with 82.4% correct classification.

Animut [17] tried to explore a semi-supervised text classification using the Amharic text documents. A total of 3,154 news articles were used to do the research. To come up with good results document preparation and preprocessing was done. Weka package is used for the classification of the preprocessed data. Machine learning techniques, Expectation maximization clustering algorithm with Naïve Bayes, Hyperpipe, and RBF Network classification algorithm were used to categorize the Amharic news items. The accuracy of the classifiers was better when the number of classes is less. The best result was obtained by the Naïve Bayes, Hyperpipe and RBF Networks classifiers with four classes (83.44 %, 82.8 and 82.4%) and the least performance is shown on the 10 categories (55.42%, 57.26% and 51.9%) respectively. The study also indicated that Naïve Bayes is more applicable to semi supervised categorization of Amharic news items.

Kemal [33] investigated the application of machine learning techniques for automatic categorization of Afan Oromo news text using Decision Tree and Support Vector Machine Classifier . Annotated news texts are used to train classifiers with six news categories: sport, business, politics, health, agriculture, and education.824 total data set of news texts were used to do the research. To come up with good result text preparation and preprocessing was done. The best result obtained by Decision Tree Classifier and Support Vector Machine is on six categories data(96.58, 84.93%) respectively. This research indicated that Decision Tree Classifier is more applicable to Afan Oromo news text than the other classifiers. The 10 fold cross validation was used for testing purposes.

Naol and Getachew [7] attempted to presents Afan Oromo text categorizations by using clustering & classification approaches. The aim of was to design, and implement Afan Oromo nonfiction text categorization model & examining the application of machine learning techniques for automatic Afan Oromo nonfiction text categorization system. Data is collected from Oromia Culture and Tourism Bureau, Oromo cultural center, online electronic documents and other nonfiction books available. Python programming language is used for tokenize, remove stop words and stem Afan Oromo nonfiction text words whereas R programming language was utilized for document indexing, Normalization, cosine similarity, and preparing documents for machine learning. Weka with java is used for splitting Afan Oromo nonfiction text document data set into train set and test set. Again Weka is used for clustering and classification of Afan Oromo nonfiction texts. By using k-mean algorithm clustering tasks were performed four times to get classes of documents. Among those, one clustering was resulted in cluster with 8 main categories were obtained as good clusters. J48, Naïve Bayes, Bayes Net, and SMO classifier algorithms were implemented for training text classification model using the 8 main classes of documents. Among those classifications algorithms, J48 algorithm shows higher performance of 94.3755% and hence it was utilized for constructing classification model. From this work they conclude that machine learning techniques can be applied for Afan Oromo nonfiction text categorization.

2.8.2. Works done in job vacancy categorization

John Lynch [20] , proposed a job title classification system using Random Forest and Support Vector Machines supervised learning algorithms. The generated prediction models make prediction based on the Top 30 most frequently occurring Job Titles Data (job descriptions) labelled with Job Titles collected from a popular national job postings website (www.irishjobs.ie). The data was collected via web scraping from www.irishjobs.ie; a popular Irish job advertisement website. This website was selected as it has a broad range of job postings. A total of 10,294 dataset is collected which are active as of 1 June 2017. The web scraping is done using the revest package on r After Several standard text-pre-processing in order to reduce dimensionality of the corpus Feature engineering was used to create a Data Model(s) of selected representative keyword generated on the basis of term frequency. The best model was the SVM linear kernel-based model, which had an Accuracy rate of 71%, Macro Average Precision of 70%, Macro Averaged Recall of 67% and a Macro Average F-Score of 66%. Whereas The least model was Random Forest based model; with a Accuracy rate of 58%, Macro Average Precision of 56%, Macro Average Recall of 55% and Macro Average F Score of 56%. At the end the researcher recommended to apply a more advanced text processing methods using NLP dictionaries, while more complex machine learning techniques can be employed such as ensemble methods to improve predictive power. And other similar and parallel research on web-based text mining, linking ends of activity to their intrinsic components may use the described method. This could be applied to domains such as marketing and medical categorization.

Amato, Flora, et al [18], proposed classification of Web Job Advertisements: A Case Study using different techniques. A total of 40,000 vacancies scraped from 12 Web sources and a subset of 412 job offers selected for being a representative sample. Then, each job vacancy has been manually labelled by domain experts at CRISP Research Centre using the qualification codes outlined in the CP2011 classifier, by looking at both offer titles and full descriptions to assign labels. This sample dataset was be used as a gold benchmark to evaluate the classification technique outcomes.

Furthermore, a common text preprocessing pipeline was used before applying any of the approaches applied that includes: tokenization, lower case reduction, html special characters substitution, stop words removal, misleading words elimination, and numbers elimination. The word stemming was performed using the Italian stemmer provided by the NLTK framework version 3.0a3 [16]. Finally, in this experimental phase only job titles have been considered as input of the classification process. Then explicit rule-based approach is applied for two goal: (i) to identify the relevant terms used in vacancy Web ads, and (ii) to relate them to the occupation codes used in the CP2011 classifier.

Then, two machine learning classifiers were used to perform the text classification purposes: the LinearSVC (an implementation of Support Vector Machine Classification using a linear kernel) and the Perceptron classifier, both built using the Scikit-learn framework. A grid search of the classier parameters maximizing the classification accuracy was performed on both classifiers. Next, LDA based approach is applied. Graph was built to compute the distance between two occupations o1, o2 as the shortest-path between them.

The result showed that the LDA approach has the lowest average shortest-path length. This means that LDA tends to classify a job offer over an occupation code that is distant averagely 1:5 from the correct classification, while rules-based approach here has the worst performance, mainly due to unclassified job offers that automatically assigns to it the maximum distance. Furthermore, if infrequent job offers neglect the (the ones occurring less than 5 time in the sample) all the distance values improve and, here, the linear SVC reaches an average distance less than 1. In other words, the SVC classier averagely classifies over an occupation code that is a one with the right one.

For supervised learning approaches the highest the number of job offers with the same code, the lowest the average distance from the gold benchmark. Differently, rule-based algorithm performs averagely well even on infrequent job offers.

The researcher suggested that to consider the full descriptions of job vacancies and to tackle the increased complexity of longer (and noisier) texts through computational linguistic approaches. (in addition to occupation codes) e.g., the required skills, contract types, business sectors, education levels, etc.

As per the review of related works, the attempts done for categorizing job vacancy announcement focused on text written in English and the local works done on text categorization focused on news domain. To fill the gap in research works and towards automating search for job vacancy announcements there is a need to conduct further study for the categorization of job vacancy announcements text written in Amharic.

CHAPTER THREE

THE AMHARIC LANGUAGE AND ITS WRITING SYSTEM

3.1. Overview

Amharic (ኣማርኛ - amarəñña) is an Ethiopian language which comes from Semitic language and used as first language by Amhara (አማራ) which is found in the northern Ethiopia [17]. The Amharic language is regarded to be the language's historical center. It is Ethiopia's (ኢትዮጵያ) working language. The majority of the 25 million or so Amharic speakers live in Ethiopia, although the language is also spoken in a number of other countries, including Eritrea (ኤርትራ), Canada, the United States, and Sweden [17] [19].

The Federal Government of Ethiopia's working language, Amharic, is spoken and written as a first or second language in many parts of the country. Amharic, like other Ethiopic script languages (Gurage, Harari, Tigre, and Tigrniya), uses fidel (ፊደል) characters, which are mostly derived from Geez [1] [17]. Around 1986, the Ethiopic script was first shown on a computer. The difficulty in computerizing the script at the time was designing a software package that could handle character design, keyboard layout, and printer setup. The work by ESTC started an enthusiastic rush to develop Ethiopic software by various IT companies and teams of individuals, resulting in a lack of uniformity on a computer around 1986. More than 35 Ethiopic software applications are currently available, each with its own character set, encoding scheme, typeface names, and keyboard layout. The recent addition of the Ethiopic range to the Unicode standard may aid in the standardization of previously incompatible applications [17].

3.2. The Amharic Characters (ፊደል)

Alphabets which is also known as characters, Fidel (ፊደል) in Amharic is a sets of letters arranged in fixed orders of the language and is used to represent a phoneme [17]. It contain consonants and vowels. The Amharic writing system consists of a core of thirty three characters each of which occur in one basic form and in six other forms called orders. The seven orders (the 1st basic form and rest six orders) of the Amharic script represent the different sounds of a consonant-vowel combination known as syllabic [4] [53]. According to [53] [4], each character describes a consonant together with its vowel, the vocalic symbol cannot be detached from the consonant element. Thus, Amharic does not use independent symbols for vowels. The non-basic forms are derived from the basic forms by more-or-less regular modifications [54]. The Amharic alphabet does not have capital and lower case distinctions. A list of the Amharic alphabets (ፊደል) with its orders is shown in appendix **I**.

3.3. Amharic Punctuation Marks

Punctuation is commonly employed in languages to create a sound gap between words or phrases. Amharic has its own set of punctuation signs, some of which are unique to the language and others which have been taken from other languages. There are approximately 17 punctuation marks in Amharic [4]. Sample of punctuation marks used in Amharic language is shown in table 3.1. The complete is listed in an appendix **II**.

Table 3. 1 Sample Punctuation Marks used in Amharic (Source: [4])

Mark	Amharic Meaning	English Meaning	Uses	
፡	ሁለት ነጥብ	Space	To separate words	Unique to Amharic Language
።	አራት ነጥብ	Full stop	To separate single phrases	
፣	ነጠላ ሰረዝ	Comma	To separate single phrases	
፤	ድርብ ሰረዝ	Semicolon	To separate more than single phrases	
?	ጥያቄ ምልክት	Question Mark	To emphasize a sentence spoken	Borrowed from Foreign Language
!	ቃለ አጋኖ	Exclamation	To give to a spoken word, phrase or sentence	
" "	ትምህርተ ጥቅስ	Double Quotation	To emphasis ones speech, says, etc	

3.4. Amharic Number System

Amharic numbering system uses Ge’ez numbering system [3]. According to [54], Amharic number characters are derived from Greek letters, and some were modified to look like Amharic fidel. In Amharic language numbers one to ten, multiples of ten (twenty to ninety), hundred, and thousand are represented by single characters. A horizontal stroke runs above and below each symbol. In the Amharic script, there is no symbol for zero. As a result, arithmetic computations utilizing the symbols are extremely complex, assuming they can even be done at all. As a result, Hindu-Arabic numerals are commonly used. Dates and page numbers in text are generally written in Ethiopic numeral systems [54]. Sample Amharic numbers are shown in appendix **III**.

3.5. Amharic word class

As other languages Amharic language has set of structural rules governing the composition of sentences, clauses, phrases, and words in a given natural language which is known as grammar (ሰዋሰዉ). According to Abeba [55], there are five word classes. These are noun, verb, adjective, adverbs and prepositions.

Amharic Nouns

Nouns are words that are used to designate a set of names, things, or locations, etc in the Amharic language [55] . Amharic nouns have the possibility to have up to two prefix and four suffixes for each stem [56]. According to [56] Amharic nouns have the following common structures and properties.

- ✓ ኦች (read as “och”) morpheme as a plural marker
- ✓ In the sentence nouns can be used as a subject.
- ✓ Nouns can also be used as object in the sentence
- ✓ And can take modifiers and quantifier

Amharic Adjectives

Adjectives in Amharic language is used to modify nouns or a pronouns by describing, identifying, or quantifying words [55] [56]. Adjectives frequently comes before noun or pronoun that they modify and gives more information about noun or pronoun it modifies. But always the words which comes before nouns can’t be an adjective. Objects can be differentiated from each other by attributes like color, shape, behavior, etc. and that differences are described by means of adjective word class.

For example, ጥቁር ጃኬት which means black jacket. In the above example the word ጥቁር/black is an adjective that modifies the noun ጃኬት/jacket. It gives more information about the color of the jacket.

Amharic Verbs

Verb is a word that indicate action [55] . It can be described as a word used to show that an action is taking place, a word to indicate the existence of a state or condition [56]. Amharic language verbs are very complex consisting of a stem and up to four prefixes and four suffixes and are inflected for person, gender, number, and time with the basic verb form being third person masculine singular. Verbs in passive voice are marked by suffixes that depend on person and number.

Amharic Adverbs

An adverb is a term that modifies the verb that follows it [56]. Adverbs in Amharic can signify time, manner, place, cause, or degree, and can also answer questions like እንዴት "how", መቼ "when", የት "where". There are only a few primitive adverbs and these are:- ገና “yet”, ክፉኛ “severely”, ቶሎ “quickly”, ጅልኛ “foolish”, etc [55] .

Preposition

Amharic prepositions are words which are usually used before nouns to show their relation to another part of a clause and they are limited in number [55] [56]. Some examples of prepositions are:- እንደ/like, ስለ/for, ከ/from, ወደ/to, etc. Amharic prepositions have meaning only when they combined with other word class. So that they are used as affixes by coming before and after words. Some examples of Amharic prepositions that come before words and after words are: - ስለ/for, እንደ/like and አጠገብ/near to , ማዶ/beyond respectively.,

3.6. Amharic Morphology

Amharic, like other Semitic languages, has a morphologically complicated structure [57]. As stated by [58] tense, aspect, and mood; different derivational categories such as passive, causative, and reciprocal; polarity affirmative/negative); relativization; and a variety of prepositions and conjunctions are all conveyed via Amharic morphemes. Verb stems in Amharic, as in other Semitic languages, are made up of a root + vowels + template merger. For example, the root verb sbr + ee + CVCVC, which leads to the stem seber ("broke") [57] [58]. A root represents a group of consonants with a common lexical meaning. This non-concatenative morphological properties makes Amharic morphology analysis more complex. In addition to this affixes also construct inflectional and derivational morphemes in Amharic. Prefix, infix, suffix, and circumfix are all examples of affixation [58].

Amharic nouns are inflected for case (i.e. accusative/ objective, possessive/genitive), number, definiteness and gender. Adjectives in Amharic can be marked for number, definiteness, cases, and gender in the same way as nouns can. Except for certain plural construction, the affixation of morphemes to convey numbers is similar to that of nouns. The verb is inflected for person, voice, tense aspect mood (TAM), number, gender and mood. As a result, a single verbal root can yield tens of thousands different verbs. Nouns in Amharic language can be derived from nouns itself, verbal roots, adjectives, stems, stem like verbs. There are only a few primary adjectives (non-derived) in the language. On the other hand many adjectives can be derived from stems, compound words, nouns and verbal roots. It can also be derived by intercalating vocalic parts into roots or adding a suffix to bound stems. Amharic verbs can also be derived from different verbal stems in many ways [57].

3.7. Problems in Amharic Writing System

There are several problems perceived in the Amharic language writing system. These problems are discussed as follows.

Existence of character variants: There are different One of the problems in Amharic writing is the redundancy of symbols used with the same pronunciation. That means there are some letters that are used for representing similar sound in Amharic [1]. Although in the Ge’ez language, these different symbols give each word different meanings, in the Amharic language they have been used interchangeably. For example, consider ሀ, ሐ and ኀ. Since all of the above three letters have different symbols, they have the same pronunciation h. Similarly both ሠ and ሰ have the same pronunciation (/s/), ዐ and አ are pronounced as /a/ and ጸ and ፀ are pronounced as (/s’/).

Spelling variation of the same word: One can imagine how the meaning of the original word is diverted to different contexts. Spelling variation of the same word: the same word is written in various forms. For example, the word ‘ሰምቶአል’ (‘he hears’) can be written in Amharic as ሰምቶአል, ሰምቷል, ሰምትዋል, etc. Spelling variation may happen also in the case of translating foreign word to Amharic. The problems resulted from use of loan words that are borrowed from other languages and that do not possess their own translation in Amharic. For instance, the word sponsor transliterated differently as ስፖንሰር or እስፖንሰር.

Formation of Compound Nouns: In Amharic language compound nouns are sometimes written as two separate words. For example, ቤተ-ሙከራ which means “laboratory” van be written as ቤተሙከራ or ቤተ ሙከራ. This happened to be inconsistent in Amharic texts and should be considered in automatic classification [54].

Inconsistency of Abbreviation: Abbreviation of concept is **another** problem that leads to inconsistency problems for automatic classification. For instance, the phrase or ዓመተ ምህረት which means AD can be written as ዓ.ም, ዓ/ም or ዓ-ም which result in an inconsistency of abbreviating Amharic words. These different representations of the same word create high dimensional vector space and it has a negative effect on the performance of learning algorithms [17]. Though, in text classification tasks such words should come into one common form.

CHAPTER FOUR

METHODS AND APPROACHES

4.1. Overview

This chapter briefly discusses the general architecture of the proposed model, data preprocessing processes and algorithm selected for categorizing job vacancy announcement text. As noted by Gonfalonieri [59], machine learning models heavily depend on data and training dataset is required in order to apply machine learning algorithms. Training data set is the actual data set used to train the model. Without a foundation of high quality training data, even the most performing algorithms can be rendered useless [60]. So it is important to give attention to the preparation of quality data [59].

Each step used for preparing data in this study is clearly discussed in the following sub-sections. First, we present the proposed architecture that describes components integrated for the purpose of categorizing job vacancy announcement. This is followed with a discussion of methods and algorithms used for data set preparation, machine learning and evaluation of the prototype.

4.2. Proposed Architecture

The proposed architecture for job vacancy announcement text categorization is shown in figure 4.1. The proposed architecture is composed of the following stages. These are document preprocessing, classification and testing of the model. Document preprocessing tasks include tokenization, normalization, removal of stop words and stemming.

Once document preprocessing task is done, the datasets is divided as training and testing data and it is converted in to an appropriate format (i.e. into the form that is suitable for machine learning algorithms). Then learning algorithms process this dataset and classify the text to its category. The performances of those classification algorithms will be evaluated by using four classification evaluation metrics, such as accuracy, recall, precision and F-measure. Finally the algorithm that performs the best is selected to construct the classification model for Amharic vacancy text and it is tested using confusion matrix.

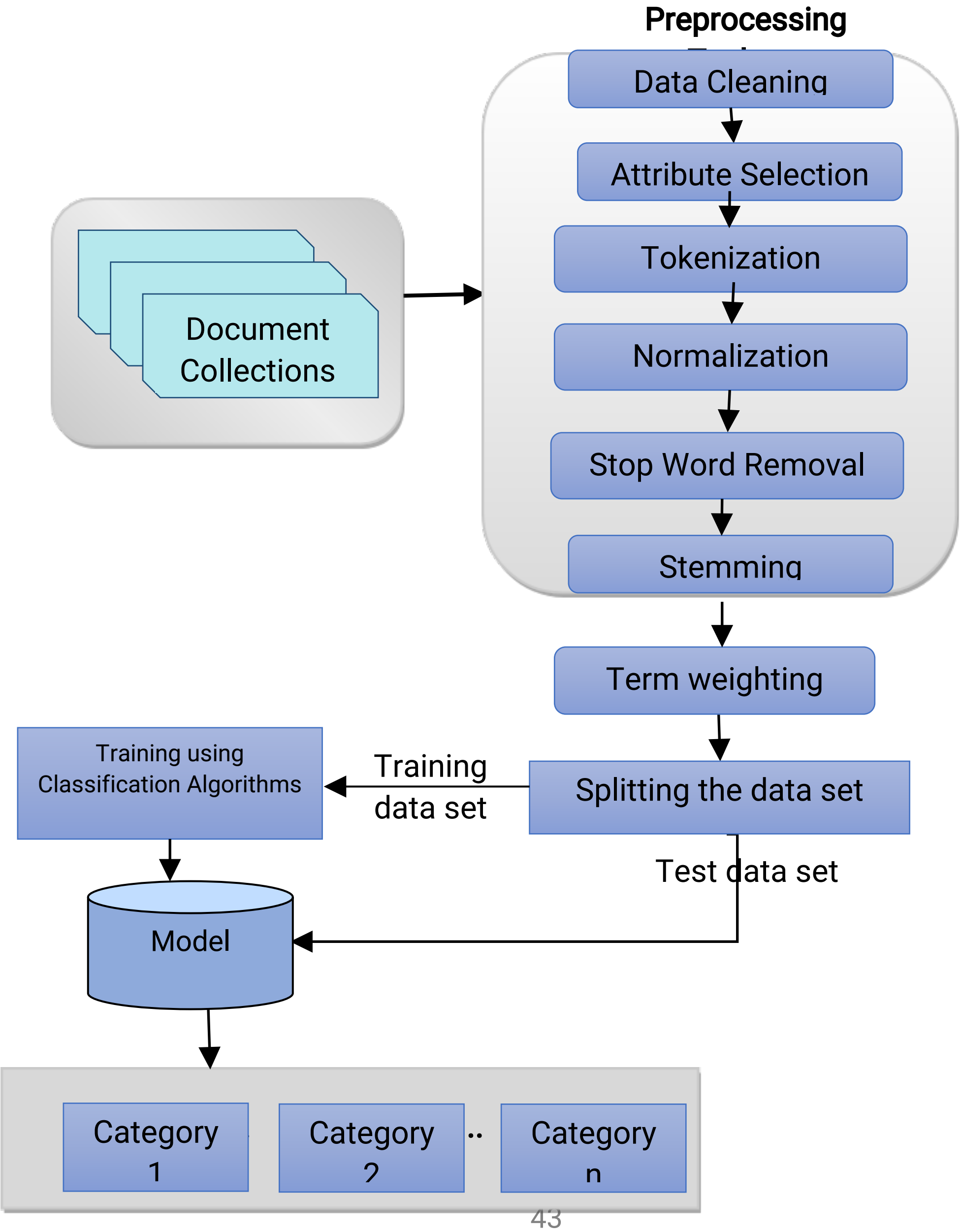


Figure 4. 1 The Proposed Architecture of Vacancy Text Categorization

4.3. Dataset Collection

The data set used for conducting this research is Amharic vacancy text which is collected from different job posting websites (https://jobwebethiopia.com, https://ethiojobs.net, https://www.ethiopianreporterjobs.com). Detail of data collection for training and testing is shown in table 4.1.

Table 4. 1 Summary of data collected from different job posting websites

Websites	Size	Percentage (%)
https://jobwebethiopia.com	614	36.5
https://www.ethiojobs.net	421	25.1
https://www.ethiopianreporterjobs.com	645	38.4
Total	1678	100

In this study, the vacancy text used to conduct the experiments is collected from the above different websites using data miner web scraping tool. Data Miner Scraper is a data extraction tool that lets users scrape any HTML web page [61] [62]. It is a browser extension software that assists you in extracting data that you see in your browser and save it into an Excel spreadsheet file. It helps to extract tables and lists from any page and upload them to Google Sheets or Microsoft Excel. With Scraper you can export web pages into XLS, CSV, XLSX or TSV files (.xls .csv .xlsx .tsv) [61] [62]. In the current research this tool is used to scrape the vacancy text from the above sources and exported as .xlsx file. Moreover, since not all contents of the document are important to categorize the document the researcher consider job requirement or educational background to classify the document as it contains complete information than job title.

One thousand six hundred seventy eight (1678) vacancy text documents were collected from the above sources for agriculture, business and economics, computer science and related fields, engineering, health, law, natural science and social science fields. Detail for the number of datasets for each categories is presented on chapter five on table 5.1.

Out of these data the researcher selected one thousand six hundred ten (1610) data to conduct the experiment. The reason why the researcher selected only 1610 from the total dataset is that; some of these data does not contain full information or some information are missing. That means there are positions that do not contain educational background like “motorist”. As educational background is considered in this study for categorization of job vacancy announcements, job positions missing educational background is removed from the dataset.

4.4. Preprocessing Vacancy Text

Online texts contain usually lots of noise and uninformative parts such as HTML tags, scripts and advertisements and on words level, many words in the text do not have an impact on the general orientation of it [63]. In order to make a document ready for model building and evaluation, preprocessing task is very important. It is required to make ready the data set used for training and testing machine learning algorithms. It is the method to improve the accuracy, efficiency, and scalability of the classification process. In order to get better experimental results, language dependent document preprocessing should be performed before automatic classification is implemented.

Text preprocessing is the task by which the text is made comfortable to the learning algorithm. In of the current study this step is applied to convert raw data in a natural language processing to the most important text features that used to identify between text-categories. In this study preprocessing of vacancy text is applied before they are used for the categorization task. The preprocessing task comprises a removal of non-informative words or characters from the text [3] [4] [5] [17]. The different tasks performed during the preprocessing phase are text cleaning, normalization, tokenization, case conversion, stop word removal, stemming and term weighting. Details of these tasks are discussed in the following subsections.

4.4.1. Data Cleaning

Data cleaning is performed before applying the text categorization process. Text data contains a lot of unnecessary tokens like digits, punctuations and symbols that should be removed before performing any further operations like tokenization, and normalizations If these tokens and characters remain in the document, then it may corrupt the document and makes the task of preprocessing challenging. The algorithm used for removing unwanted characters works as follows. Special characters and numbers list for example 1, 2, 3…/,::, ፤, ፣, ?, ... is prepared and tokenized words are checked whether it in this list or not. If the token is in special word list it is removed; it will remain otherwise. The algorithm used for removing unwanted characters and digits is shown below in Figure 4.2.

```
Algorithm RemoveUnwantedCharachtersAndDigits()
    Open file
    Open list of special characters
    While not end of file
        If a character is in list of special characters or number then
            Remove character
        End if
    End while
    Return cleaned text
End Algorithm
```

Figure 4. 2 An Algorithm to Remove Unwanted Characters and Digits

There are some vacancy texts written in English language. The researcher used Google Translator API to translate these texts written in English into Amharic. Since there are few number of abbreviations in the data these are expanded manually.

4.4.2. Attribute Selection

Vacancy text can be categorized based on different parameters like job items, locations, experience, level of education, employment type and educational background on each job posting website. Among these parameters the researcher selected to categorize the vacancy text based on their items or categories using educational requirement. Most researches done on vacancy announcement had used job titles to categorize vacancy texts into their predefined categories. But job title doesn't give full information for classification [18].

Since the vacancy text used in this research is taken from the websites that categorize each job announcement by their educational background. As a result these job items that placed under one category is taken as instances for that category. For example the class label for these job items under engineering are "engineering" and the same is true for other categories.

Since most vacancies announced as; for instance, information technology, computer science, information systems and related fields, we have categorized related fields as a single category. Categorizing these related fields into a single category is used to minimize the redundancy of text in the dataset. The job announcement that does not contain educational background is not considered in this study, where 1610 dataset with eight categories are selected to conduct the experiment. The detailed categories used will be discussed under chapter five, section 5.2.

4.4.3. Tokenization

For many natural language processing tasks, we need to first access each word in the string. So to access each word, the text should have to be broken into smaller components. Tokenization is the text preprocessing task of breaking up text into smaller components of text known as tokens which are bases for document representation [5] [64]. Words, numbers, punctuation marks, and others can be considered as tokens [65]. Further processing is generally performed after a piece of text has been appropriately tokenized.

Token is a single entity that is a building blocks for sentence or paragraph [66]. Sentence tokenizer algorithm breaks text paragraph into sentences and Word tokenizer breaks text paragraph into words [66]. Sentence tokenizer algorithm reads the cleaned text file line by line upto the end of file and breaks the words in each line in to tokens by using spaces between words and characters. The algorithm for word tokenization is shown i

```
Algorithm tokenization ()
        Open file
        Read the content of the file line by line
        While not end of file
                Split into word by space
        End while
        Return list of tokenized words
End Algorithm
```

Figure 4. 4 An Algorithm to Tokenize Vacancy Text

4.4.4. Normalization

In Amharic language writing system there are some characters with the same sound but have different symbols. These different symbols must be considered as similar because they do not have effect on meaning [17]. As a result, in this study, all different symbols of the same sound are converted into one common character. Thus, for example, if the character was one of ሐ፣ሓ፣ሃ፣ኀ or ኃ (all of them with a similar sound , h) then it was converted to ሀ. By the same token, all orders of ሠ (with the sound s) were changed to their equivalent respective orders of ሰ, all orders of ዐ (with the sound a) were changed to their equivalent respective orders of አ, all orders of ፀ (with the sound tse) were changed to their equivalent respective orders of ጸ. The algorithm for normalization works as follows. List of similar characters to be normalized with its corresponding selected replacement character is prepare then the algorithm checks for characters with the same sound in the tokenzed words and replace it with the corresponding replacement character which is prepared list. The algorithm is shown in

Figure 4.3.

```
Algorithm convertIntoOneCommonCharacter ()
    Open file
    Open list of similar characters
    While not end of file
        If character in list of similar characters
            Replace characters with selected character
        End if
    End while
        return list of converted words
End algorithm
```

Figure 4. 3 An Algorithm to normalize Amharic Characters

4.4.5. Stop Word Removal

Stop words are the most common words in any natural language. For the purpose of analyzing text data and building categorization model, stop words might not add much value to the meaning of the document since they carry no predictive influence of the model and can also be removed [67]. Removing stop words is not a hard and fast rule in NLP. It depends upon the task that we are working on. For tasks like text categorization, where the text is to be classified into different categories, stop words are removed or excluded from the given text so that more focus can be given to those words which define the meaning of the text. However, in tasks like machine translation and text summarization, removing stop words is not advisable [68]. The benefits of removing stop words include the following [68]. First, it decreases dataset size and the time to train the model. It also helps to improve the performance of the classifier as there are fewer and only meaningful tokens left. The algorithm used to remove stop words work as follows. List of common Amharic stop words and stop words considered in this research is prepared and saved as separate file. Then the algorithm read vacancy text and check if the word is in the list of stop words. If the word is found in the list of stop words; the word is removed. Otherwise the text is moved to the non-stop words. Figure 4.5 presents the algorithm used for stop word removal from the data set.

```
Algorithm stopWordRemoval()
    Open file
    Open list of Amharic stopwords
    While not end of file
        If a word exist in stop word list then
            Remove a word
        else
            Move to non-stop word list
        end if
    end while
    return non-stop word lists
End Algorithm
```

Figure 4. 5 An Algorithm for Stop Word Removal from Vacancy Text

4.4.6. Stemming

Stemming is a process of linguistic normalization, which reduces words to their word root or chops off the derivational affixes of a word. For example, Amharic words, "ማህበራዊ"; "በማህበር" are reduced to the stem word "ማህበር" [66]. Stemming words helps to define words in the same context with the same term and this consequently reduce dimension of the word in the training corpus [3]. Stemming programs are commonly referred to as stemming algorithms or stemmers. Stemming is an important part of the pipelining process in Natural language processing. Tokenized words are given as an input to the stemmer [69]. Even if there are different famous stemming algorithms, such as porter stemmer for English, they cannot be directly applied for local languages. As a result, the researcher identified the prefix and suffixes in Amharic [4], and wrote a code by using python programming language to remove these affixes. The algorithm used to stem Amharic vacancy words work as follows. List of prefixes, suffixes and exceptional words were prepared. Exceptional words list are list of words with affixes which may change the meaning of the word or reduce the word to meaningless if removed. Consider the word "ከተማ" which means city. Since "ከ" is in prefix list it is going to be removed and the rest word remains meaningless. So ከ in this and such words should not be removed and "ከተማ" is in exceptional word list. After opening Amharic vacancy words the algorithm check whether the word start with prefix or not. If word start with prefix and not in exceptional word list its prefix is removed. Otherwise the word remain as it is. Similarly if word ends with suffix and not in exceptional word list its suffix is removed. Otherwise the word remain as it is. Figure 4.6 presents the algorithm used for stemming vacancy text from the data set.

```
Algorithm stemming ()
    Open file with list of words
    While not end of file
        If word start with prefix
            If word not in exceptional list
                Remove prefix
            End if
        End if
        If word ends with suffix
            If word is in exceptional word list
```

Figure 4. 6 An algorithm for Stemming Amharic Vacancy Text

4.5. Feature Extraction using Term Weighting

After the word stems are generated, the next step is feature extraction. The relevance of a word to the topic of a document is measured using term weighting. These terms are identified from training data such that each class can be represented with the appropriate terms (class representatives) of that class. So the term weighting measures the importance of the term to represent the given documents and is proportional to the number of times term appears in the document. Here, the document, D is represented in a vector space as follows.

$D= (w_1, w_2, w_3,...w_{|T|})$

Where w_i is the weight of i^{th} term in document D and |T| is the total number of unique terms in the document collection. There are different term weighting approaches and most of them are based on the following characteristics [3].

- The importance of a word to the given documents is proportional to the number of times it exists in the documents.
- If the word appears in most of the documents, its discriminating power between

documents is less.

The most common term weighting approaches used in text categorization is term frequency × inverse document frequency (TFIDF) weighting [3] [5] [4]. It is calculated by multiplying term frequency by inverse document frequency.

Term frequency (TF) is a technique for weighting by counting the occurrences of terms, t within a document, d normalized by the amount of words within a document.

$$\text{TF (t, d)} = \frac{\text{Number of times term t appers in a document}}{\text{Total number of terms in the document}}$$

(4. 1)

Inverse Document frequency (IDF) is used for finding the importance of the word for representing a document. It is based on the fact that less frequent words are more informative and important. IDF is computed by the formula:

$$\textbf{IDF=log}\,\frac{\mathbf{N}}{\mathbf{n}} \qquad \textbf{(4. 2)}$$

Where, N is the total number of documents in the collection and n is the number of documents containing a word.

TF*IDF is a numeric measure that is used to score the importance of a word in a document based on how often did it appears in document d and across a given collection of documents, D.

$$\ast\,(\quad , d, \quad) = (\quad , d) \times (\quad , \quad) \qquad \textbf{(4. 3)}$$

Where w denotes the term; d denotes each document; D denotes the total collection of documents.

To generate the document vectors for conducting the experiment we used TF*IDF weighting approach. And finally a word document matrix is constructed , that shows the frequency of words that occur in a collection of documents.

4.6. Machine Learning Algorithm

Automatic classification of documents using machine learning approach requires the learning process to be initiated by supplying the examples labeled with their class-category from which the systems starts to learn. According to Sebastiani [26], the essential idea is to infer a classifier (i.e. a rule that decides whether or not a document should be assigned to a category) from a set of labeled documents (i.e. documents with known category assignments). In this study three popular supervised learning algorithms are used to classify vacancy text, which are Naïve Bayes classifier, SVM (Support Vector machine) and KNN (K-Nearest Neighbor).

Naïve Bayes classifier uses the joint probabilities of words co-occurring in the category training set and the document to be classified to calculate the probability that the document belongs to each category. The document is assigned to the most probable category. The naïve assumption in this method is the independence of all the joint probabilities [4].

NB classifier works on conditional probability. Conditional probability is the probability that something will happen, given that something else has already occurred. By using conditional probability, it is possible to calculate the probability of an event using its prior knowledge. Naive Bayes classifier calculates the probability of an event in the following steps: [70]

In case of a single feature

- Calculate the prior probability for given class labels
- Find Likelihood probability with each attribute for each class
- Put these value in Bayes Formula and calculate posterior probability.
- See which class has a higher probability, given the input belongs to the higher probability class.

In case of a multiple feature

- Calculate prior probability for given class labels

- Calculate conditional probability with each attribute for each class
- Multiply same class conditional probability
- Multiply prior probability with step 3 probability
- See which class has higher probability, higher probability class belongs to given input set step.

Support Vector Machines represents every document as a vector and tries to find a boundary that achieves the best separation between the groups of vectors. The system is trained using positive and negative examples of each category and the boundaries between the categories are calculated. A new document is categorized by calculating its vector and determining the partition of the space to which the vector belongs.

SVM algorithm works by making a straight line between two points. All of the data points on one side of the line represents a category and the data points on the other side of the line placed into a different category where there can be an infinite number of lines to choose from [71]. SVM classifier is better than other algorithms, like k-nearest neighbors, in that it chooses the best line to classify your data points. It chooses the line that separates the data and is the furthest away from the closet data points as much as possible.

KNN is an effective and powerful classification and regression algorithm because it does not assume anything about the data, other than a distance measure can be calculated consistently between two instances. As such, it is called non-parametric or non-linear as it does not assume a functional form [28].

KNN classifier is to predict the target label by finding the nearest neighbor class. The closest class to the point which is to be classified is calculated using Euclidean distance. The algorithm works as follows: First, given a test document x, the KNN among the training documents are found. The category labels of these neighbors are used to estimate the category of the test document. In the traditional approach, the most common category label among the k-nearest neighbors is assigned to the test document [17].

KNN algorithms decide a number k which is the nearest Neighbor to that data point which is to be classified. For example, if the value of k is 5 it will look for 5 nearest Neighbors to that data point. Deciding the k can be the most critical part of K-nearest Neighbors. If the value of k is small then noise will have a higher dependency on the result. Overfitting of the model is very high in such cases and bigger the value of K will destroy the principle behind KNN. So the optimal value of K can be found by using cross -validation. The following algorithm shows how KNN algorithm works [72]:

- Select the number K of the neighbors
- Calculate the Euclidean distance of K number of neighbors
- Take the K nearest neighbors as per the calculated Euclidean distance.
- Among these k neighbors, count the number of the data points in each category.
- Assign the new data points to that category for which the number of the neighbor is maximum.
- Our model is ready.

4.7. Evaluation Techniques

In the current study the performance of the classification model is evaluated by using four indexes. They are Accuracy, Precision, Recall and F1-score. The common way for computing these indexes is based on the confusion matrix as shown below [73]:

Table 4. 2 Confusion Matrix

		Predicted Class	
Actual Class		**Positive**	**Negative**
	Positive	**True Positive**	**False Negative**
	Negative	**False Positive**	**False Positive**

The confusion matrix shows the classification result in terms of true positive, rue negative, false positive and false negative.

- **True Positive:** is an outcome where the model correctly predicts the positive class.
- T**rue negative:** is an outcome where the model correctly predicts the negative class.

- **False positive:** is an outcome where the model incorrectly predicts the positive class.
- **False negative:** is an outcome where the model incorrectly predicts the negative class.

Accuracy: is the proportion of true results among the total number of cases examined.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \qquad (4.\ 4)$$

***Pr*ecision**: is the ratio of correctly predicted positive items to the total predicted positive items.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (4.\ 5)$$

Recall: quantifies the number of positive class predictions made out of all positive examples in the dataset.

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (4.\ 6)$$

F1 score: is a number between 0 and 1 and is the harmonic mean of precision and recall.

$$F1 = 2 * \frac{\text{Precision*Recall}}{\text{Precision+Recall}} \qquad (4.7)$$

CHAPTER FIVE

EXPERIMENTATION AND DISCUSSION

5.1. Overview

In this chapter the details of all experimentations and discussion of the result have been presented. The experimentation is conducted to see the steps and methods we followed works in classifying job vacancy announcement texts into their predefined classes, such as “ጤና” (health) “ምህንድስና” (engineering), “የኮምፒዉተር ሳይንስ ዘርፎች” (computing), “ተፈጥሮ ሳይንስ” (natural science) “ማህበራዊ ሳይንስ” (social science), “ህግ” (law), “ግብርና” (agriculture) and “ቢዝነስ እና ኢኮኖሚክስ” (business and economics).

As discussed above under chapter four each items/data used is taken from the website which has a category based on job titles and educational background. The researcher makes little modification by combining these items which ask for related fields like computer science, information technology, and information system into one category. For the classification task, three algorithms are used, such as Support Vector Machine, Naïve Bayes and K-Nearest Neighbor algorithms. The result obtained from the experimentation is discussed and a comparison of these classification algorithms is done to select the best classification algorithm among the above classifiers.

5.2. Data Set Preparation

For conducting the experimentation data was extracted from https://www.jobwebethiopia.com, https://www.ethiojobs.net, https://www.ethiopianreporterjobs.com. using data miner tool. A total of 8 categories and 1610 documents were used in the experimentation process. All documents used for conducting the experiments are labeled in consultation with domain experts in to eight classes: “ጤና” (health) “ምህንድስና” (engineering), “የኮምፒዉተር ሳይንስ ዘርፎች” (computing), “ተፈጥሮ ሳይንስ” (natural science) “ማህበራዊ ሳይንስ” (social science), “ህግ” (law), “ግብርና” (agriculture) and “ቢዝነስ እና ኢኮኖሚክስ” (business and economics). In order to find how important a word in job vacancy text is in comparison to other words, TF-IDF term weighting technique was used to vectorize words.

For conducting the experiments the documents were divided into training and test data sets using percentage split. So that from the total of documents 80% of them were used to train the model, and 20% of them were randomly selected for testing. Accordingly, 1288 data set was used for training the classifier and 322 for testing. Many researchers in text classification used 20% of dataset for testing and achieved best accuracy [2]. The performance of all the algorithms (such as NB, SVM and KNN) employed in the current study were evaluated using the test data individually by using confusion matrix. A total of three experiments were conducted with these algorithms. All experiments have been done in Python programming language (Sklearn) which is popular for NLP and text processing. The results of each algorithm and finding of the research are discussed in the next section. The number of job items by categories is shown in table 5.1.

Table 5. 1 Job Items with their Corresponding Numbers

No.	Category Name	Data used in each category	
		Count	**%**
1.	ግብርና/Agriculture	120	7.5%
2.	ቢዝነስ እና ኢኮኖሚክስ /Business and Economics	234	14.6%
3.	ምህንድስና/ Engineering	272	17%
4.	ኮምፒዉቲንግ/ Computing	213	13.3%
5.	ጤና/Health	307	19.1%
6.	ተፈጥሮ ሳይንስ/ Natural Science	184	11.5%
7.	ህግ/ Law	109	6.8%
8.	ማህበራዊ ሳይንስ/ Social Science	170	10.2%

Table 5. 2 Sample Data Set Used for Experimentation

Educational Background	Category
በሲቪል ምህንድስና ወይም በሌላ በማንኛውም ተዛማጅ መስኮች የመጀመሪያ ዲግሪ።	Engineering
ሁለተኛ ዲግሪ / የመጀመሪያ ዲግሪ በነርስ	Health
የዩኒቨርሲቲ ዲግሪ በአመጋገብ ወይም በሕዝብ ጤና	Health
የመጀመሪያ ዲግሪውን በጋዜጠኝነት እና በኮሙዩኒኬሽን ፣ በፖለቲካ ሳይንስ እና በዓለም አቀፍ ግንኙነት ፣ በፌዴራሊዝም ፣ በቋንቋና ሥነ ጽሑፍ ትምህርት	Social
ዲግሪ በጋዜጠኝነት እና በኮሙዩኒኬሽን ፣ በቋንቋ ፣ በስነ-ጽሑፍ እና ውድድር	Social
የመጀመሪያ ዲግሪ በግዥ እና አቅርቦት አስተዳደር ፣ አስተዳደር ፣ ኢኮኖሚክስ ፣ ቢዝነስ አስተዳደር ወይም ተዛማጅ የጥናት መስክ	Business
ዲግሪ በነርስ ወይም በሕዝብ ጤና ላይ	Health
የመጀመሪያ ዲግሪ/ ኮሌጅ ዲፕሎማ በአዋላጅ እና / ወይም በሌሎች የጤና ነክ መስኮች ።	Health
በነርስ ወይም በሕዝብ ጤና ውስጥ የሳይንስ / ዲፕሎማ የመጀመሪያ ዲግሪ	Health
በኮምፒተር ሳይንስ በኤሌክትሪክ ምህንድስና ፣ በሃርድዌር ምህንድስና ወይም በተዛማጅ ዘርፎች ሳይንስ ዲግሪ	Computing
ዲግሪ ወይም ማስተርስ / ግብርና ፣ ማህበራዊ ሳይንስ ፣ ማህበረሰብ ልማት ወይም ሌላ አግባብነት ያለው መስክ	Agriculture
በተፈጥሮ ሳይንስ (ባዮሎጂ እና ኬሚስትሪ) መምህር የሳይንስ ባችለር	Natural
በተፈጥሮ ሳይንስ ወይም በተዛማጅ የመጀመሪያ ዲግሪ	Natural
ዲግሪ ወይም በባዮሎጂ ፣ ባዮኬሚስትሪ ፣ ፋርማኮሎጂ	Natural
በባዮሎጂ የመጀመሪያ ዲግሪ ፣ በሞለኪዩላር ባዮሎጂ ውስጥ ጠንካራ ዳራ (ሳይንሳዊ እና ቴክኒካዊ) ።	Natural
የመጀመሪያ ዲግሪ በግብርና ፣ በአርብቶ አደር ልማት ፣ በአካባቢ ሳይንስ ወይም በተዛማጅ መስክ።	Agriculture
የመጀመሪያ ዲግሪ በግብርና ፣ በአርብቶ አደር ልማት ፣ በአካባቢ ሳይንስ ወይም በተዛማጅ መስክ	Agriculture
የሳይንስ የተፈጥሮ ሀብት አስተዳደር ወይም ተዛማጅ የጥናት መስክ የዩኒቨርሲቲ ዲግሪ	Agriculture

5.3. Classification result

In this study three supervised machine learning classifiers, such as support vector machine, k Nearest Neighbor and Naïve Bayes classifiers are used to categorize the vacancy text.

5.3.1. Experimental set up

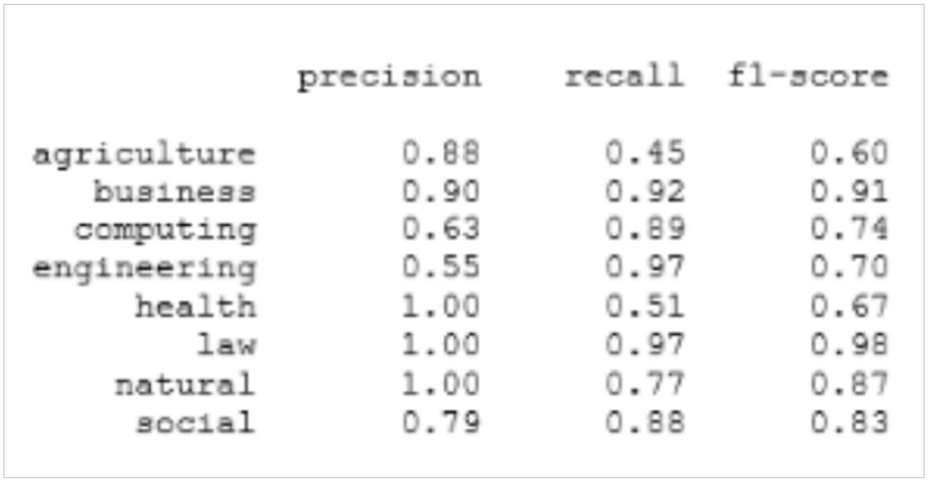
Three experiments have been conducted in this study. The data used to conduct the experiments contains two attributes, text data (vacancy text) and the class label of the vacancy text, which are “ጤና” (health)   “ምህንድስና” (engineering), “የኮምፒዉተር ሳይንስ ዘርፎች” (computing), “ተፈጥሮ ሳይንስ” (natural science) “ማህበራዊ ሳይንስ” (social science), “ህግ” (law), “ግብርና” (agriculture) and “ቢዝነስ እና ኢኮኖሚክስ” (business and economics).

In order to find how important a word in document is in comparison to the corpus, words are vectorized by using TF-IDF weighting technique. In order to examine the applicability of machine learning algorithm to categorize the vacancy text, Naive Bayes, Support Vector Machine and K-Nearest Neighbor compared with the same data set and categories. For conducting the experiments the documents were divided into training and test data sets using percentage split. So that from the total of documents 80% of them were used to train the model, and 20% of them were randomly selected for testing.

5.3.2. Experimental result

Classification Using Support Vector Machine

The first experiment was conducted by using support Vector Machine. The most widely used library for implementing machine learning algorithms in Python is scikit-learn. Scikit-learn svm.SVC() is the class used for SVM classification. The parameters used to implement SVM in this study are C which is the regularization parameter, C, of the error term, kernel type, degree of polynomial kernel and coefficient (gamma) for kernel is set first and then the model is created. The accuracy of SVM classifier is shown in table 5.3.



```
             precision    recall  f1-score

agriculture       0.88      0.45      0.60
   business       0.90      0.92      0.91
  computing       0.63      0.89      0.74
engineering       0.55      0.97      0.70
     health       1.00      0.51      0.67
        law       1.00      0.97      0.98
    natural       1.00      0.77      0.87
     social       0.79      0.88      0.83
```

Figure 5. 1 Accuracy of Support Vector Machine Algorithm

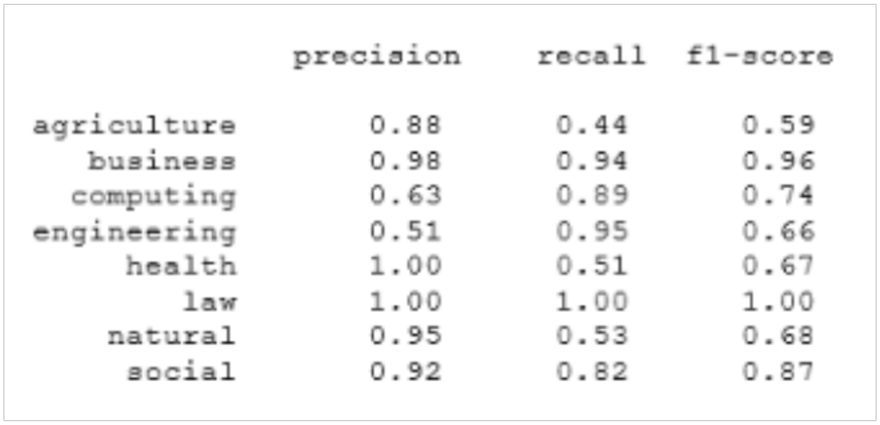
SVM achieved an accuracy of 76.4%. In law category, SVM achieved a precision of 100% and recall of 97% and the F-measure of the 98%. For business and economics category, SVM achieved a precision of 90% and recall of 92%. The F-measure of the SVM classifier for the category business and economics is 91%. In natural science category, SVM achieved a precision of 100% and recall of 77%. The F-measure of the SVM classifier for the category natural science is 87%.

On the other hand social science category achieved a precision of 79% likely to be correct and 88% of recall. The F-measure of the SVM classifier for the class social science is 83%. For computing category, SVM achieved a precision of 63% and recall of 89%. The F-measure of the SVM classifier for the category computing is 74%.

For engineering category, SVM achieved a precision of 55% and recall of 97%. The F-measure of the SVM classifier for the category engineering is 70%. The performance of SVM classifier for health category was 100% precision and 51% recall and F-measure of 67%. As we have observed from the above table the precision and recall for agriculture category are 88% and 45% respectively. The F-measure of SVM classifier for agriculture shows 60%.

Classification Using K-Nearest Neighbor

The second experiment was conducted by using K-Nearest Neighbor. Classifier implementing the k-nearest vote. The parameters include n-neighbor (default=5), weights (default=uniform), algorithm to compute the nearest neighbor, leaf_size (default=30), metric, metric_params and n_jobs (default=1). The accuracy of KNN classifier is shown in figure 5.4



```
             precision    recall  f1-score

agriculture       0.88      0.44      0.59
   business       0.98      0.94      0.96
  computing       0.63      0.89      0.74
engineering       0.51      0.95      0.66
     health       1.00      0.51      0.67
        law       1.00      1.00      1.00
    natural       0.95      0.53      0.68
     social       0.92      0.82      0.87
```

Figure 5. 2 Accuracy of K-Nearest Neighbor Algorithm

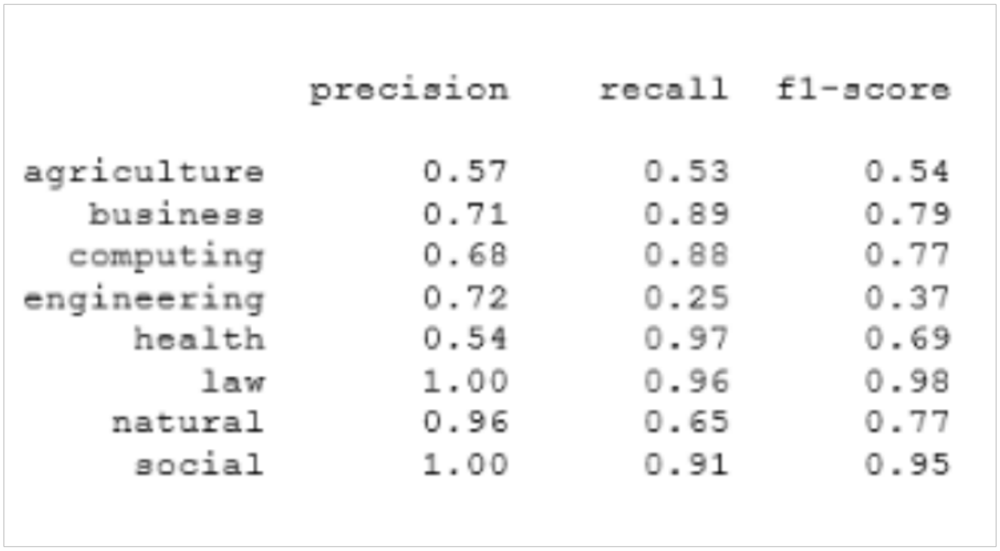
K-Nearest Neighbor achieved an accuracy of 75.5%. In law category, KNN achieved a precision of 100% and recall of 100% and the F-measure of the 100%. In business and economics category, KNN achieved a precision of 98% and recall of 94%. The F-measure of the KNN classifier for the category business and economics is 96%. On the other hand social science category achieved a precision of 92% likely to be correct. And a recall of 82%. The F-measure of the KNN classifier for the class social science is 87%.

For computing category, KNN achieved a precision of 63% and recall of 89%. The F-measure of the KNN classifier for the category computing is 74%. In natural science category, KNN achieved a precision of 95% and recall of 53%. The F-measure of the KNN classifier for the category natural science is 68%. The performance of KNN classifier for health category was 100% precision and 51% recall and F-measure of 67%.

For engineering category, SVM achieved a precision of 51% and recall of 95%. The F-measure of the KNN classifier for the category engineering is 66%. As we have seen from the above table the precision and recall for agriculture category are 88% and 44% respectively. The F-measure of SVM classifier for agriculture shows 59%.

Classification Using Naïve Bayes

The third experiment was conducted by using Naïve Bayes. The Multinomial NB classifier is suitable for text classification where it assumes that features are drawn from simple multinomial distribution. The default parameters are alpha (default=1.0), fit_priori (default=True) and class_priori (default=none).The accuracy of NB classifier is shown in figure 5.5



```
             precision    recall  f1-score

agriculture       0.57      0.53      0.54
   business       0.71      0.89      0.79
  computing       0.68      0.88      0.77
engineering       0.72      0.25      0.37
     health       0.54      0.97      0.69
        law       1.00      0.96      0.98
    natural       0.96      0.65      0.77
     social       1.00      0.91      0.95
```

Figure 5. 3 Accuracy of Naive Bayes Algorithm

Naive Bayes achieved an accuracy of 70.2%. In law category, NB achieved a precision of 100% and recall of 96% and the F-measure of 98%. In social science category, NB achieved a precision of 100% and recall of 91%. The F-measure of the classifier for the category social science is 95%.

In business and economics category, KNN achieved a precision of 71% and recall of 89%. The F-measure of the KNN classifier for the category business and economics is 79%.

For computing category, NB achieved a precision of 68% and recall of 88%. The F-measure of the NB classifier for the category computing is 77%. Similarly natural science category achieved a precision of 96% likely to be correct. 4% of them were incorrectly classified in other categories. The recall in this class is 65%. The F-measure of the NB classifier for the class natural science is 83%.

The performance of NB classifier for health category was 54% precision and 97% recall and F-measure of 69%. As shown on the above table the precision and recall for agriculture category are 57% and 53% respectively. The F-measure of SVM classifier for agriculture shows 54%.For engineering category, NB achieved a precision of 72% and recall of 25%. The F-measure of the NB classifier for the category engineering is 37%.

Table 5. 3Summary of the result obtained for the alternate experiments conducted using each classification algorithm

	Accuracy	Average Precision	Average Recall	Average F-measure
SVM	76.4%	84.375%	79.625%	81.75%
KNN	75.5%	86.875%	71.75%	76.375%
NB	70.2%	77.3%	75.5%	76.38%

The performance of these classification algorithms in terms of accuracy, precision, recall and f-measure are shown below in figure 5.1

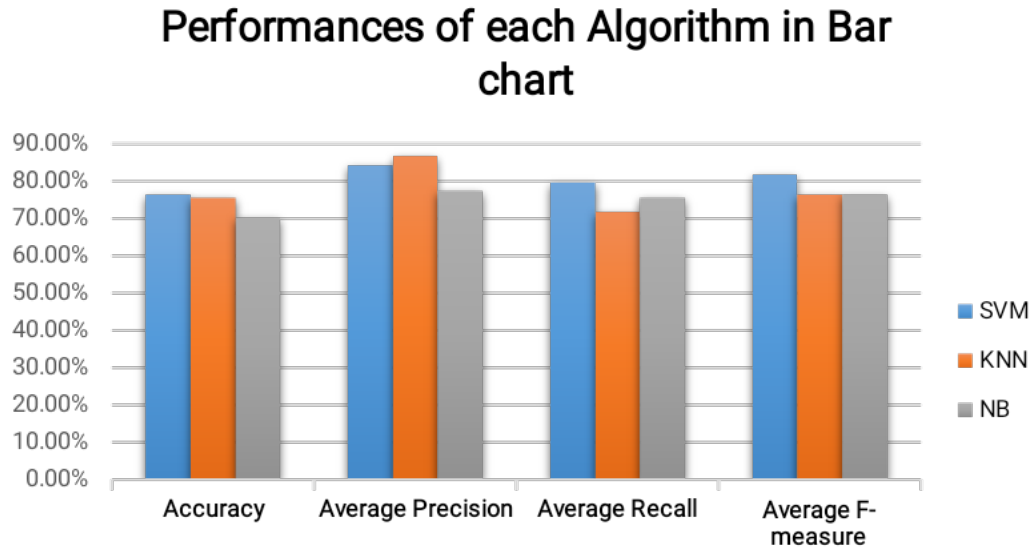


Figure 5. 4 Performances of each Algorithm by using Bar Chart

As it is observed both from table 5.6 and figure 5.1 in terms of accuracy SVM algorithm outperforms 76.4% than KNN and NB algorithms. When we see in terms of precision KNN classifier achieved the higher performance of 86.875% average precision followed by SVM and NB classifiers respectively. Also when compared with average recall each algorithm performed SVM algorithm achieved the best average recall of 79.625% followed by NB and KNN algorithms respectively. On the other hand SVM with regard to F-Measure SVM algorithm achieved the best performance of 81.75% than KNN and NB algorithms. Hence, based on its performance, the SVM classifier is selected to construct the classification model of Amharic job vacancy announcement text in this study.

Table 5. 4 Confusion matrix of the SVM algorithm

Predicted

Actual

Job vacancy type	ግብርና	ቢዝነስ እና ኢኮኖሚክስ	ኮምፒዉቲንግ	ምህንድስና	ጤና	ህግ	ተፈጥሮ ሳይንስ	ማህበራዊ ሳይንስ	**Total**
ግብርና	15	4	0	5	2	0	0	1	27
ቢዝነስ እና ኢኮኖሚክስ	0	46	0	4	0	0	0	0	50
ኮምፒዉቲንግ	0	0	33	3	0	0	0	1	37
ምህንድስና	0	0	2	57	0	0	0	0	59
ጤና	0	0	3	22	31	0	1	4	61
ህግ	0	0	0	0	0	24	0	1	25
ተፈጥሮ ሳይንስ	0	0	3	4	0	0	26	0	33
ማህበራዊ ሳይንስ	0	0	0	4	2	0	0	24	30

5.4. Discussion of the Result

In the current study, three supervised machine learning algorithms are used to categorize vacancy text document. Support Vector Machine, K-Nearest Neighbor and Naïve Bayes are the algorithms tested to categorize the text document. As shown above the experimental result of each algorithm has achieved different performance. The comparison of all three algorithm results in terms of their accuracy is shown above in table 5.6

As it is observed from the experimental result, each algorithm registered different performance to categorize Amharic vacancy text. In terms of accuracy, Support Vector Machine is the best algorithm that achieved the highest performance of 76.4% as compared to the algorithms used in the current study.

The “Law” category registered high classification result of 100%, 97%, and 98% F-Measure in SVM, KNN and NB classifier respectively when compared to category like agriculture, business and economics, computing, engineering, health, natural science and social science. As it is shown in table 5.7, given a total of 30 ህግ (law) categories in test data; out of the given law categories 29 of them were classified as law and the rest one law category labeled as social science. That is because of there are few vacancy text that share the same or common words compared to other categories used in this study. For example, consider the sentence የመጀመሪያ ዲግሪ በኤሌክትሪካ ምህንድስና፣ በኮምፒዉተር ምህድስና፣ በኮምፒዉተር ሳይንስ ወይም ተዛማጅ መስኮች which means “first degree in electrical engineering, computer engineering, computer science or related fields”. In this example the word engineering, computer and science are used repetitively in different categories like engineering, computing, natural science, social science and other categories and less with law category. This condition affects the performance of categories that share these common terms where law is unique from others comparatively. So that we can say law category share the least common words with the other else categories. That is the reason why law category achieved better performance.

On the other hand, “agriculture” category has the least performance in each algorithm where out of 27 given test data 15 of them correctly classified as agriculture, 4 of them categorized as business and economics, 5 of them categorized as engineering, 5 of them categorized as engineering and 1 of them categorized as health. A discussed above the agriculture category share common words with other categories and these highly affect the performance of the classification in this category.

The result obtained is promising result to design vacancy text categorization model for jobs announced in Amharic language. There are different challenges faced in this study. The first one is there are job positions that allow different fields of study. For example, የመጀመሪያ ዲግሪ በጤና ዘርፎች፣ በንግድ አስተዳደር፣ በግብይት ወይም ተዛማጅ መስኮች which means “first degree in health fields, business administration, marketing or related fields”. So, the presence of these different fields of study in a single document can challenge the machine and affect the performance of classification. The other challenge is as discussed above there are categories that share the common terms and found in both categories. This is another problem that affect the performance of the classifier and accuracy of categories.

CHAPTER SIX

CONCLUSION AND RECCOMENDATION

6.1. Conclusion

Nowadays, large amount of vacancy announcement has been uploaded and generated on the web daily. This increases the amount of text that are available on the web in electronic form which is difficult to organize and manage manually. In the current system the concerned body prepare the content of the job, such as ID of the job, title of the job, publishing date, job descriptions, salary and benefits, educational background, work experience and related information and categorize based on its ID, title, experience (and/or based on the rule) that the expert defined and saves it accordingly, so that it can be retrieved later by its ID, title, date, experience and category when needed. In the last few years, automatic text classification systems have proven to be just as accurate, correctly categorizing over 90% of the text classification. Since manual categorization is based on human judgments; it is accurate. But it is time consuming and inconsistent. So there has been a switch from manual to automated systems

The focus of this study is therefore to apply machine learning techniques on job domains. The data used in this study is extracted from web posting websites (https://jobwebethiopia.com, https://ethiojobs.net, https://www.ethiopianreporterjobs.com) using a data miner tool and saved as .xlsx file. Then preprocessing tasks such as data cleaning, normalization, tokenization, stop word removal and stemming were applied using python programming language to clean and make ready the data set for machine learning algorithms. In order to find how important a word in document is in comparison to the corpus, words are vectorized by using TF-IDF weighting technique. Three machine learning algorithms (i.e. Support Vector Machine, K-Nearest Neighbor and Naïve Bayes) are tested for categorizing vacancy text into its category or items.

The experimental result shows that Support Vector Machine algorithm outperforms K-Nearest Neighbor and Naïve Bayes with an accuracy of 76.3% and hence the model constructed by SVM is selected for Amharic job vacancy categorization task.

The SVM model works well for the "law" category because it is the category that share the least common terms compared to other categories; where as "agriculture" is the category that has the least performance. The factors that affect the performance of classification algorithms include presence of different field of studies in a single document and there are categories that share common terms and found in both categories which is highly affect the categorization performance.

6.2. Recommendation

Based on the findings of the study, the following recommendations are forwarded for conducting further future research.

- There are conflicting tags assigned to the data set as a result of use of similar words in different categories which needs to analyse texts semantically. It is therefore recommended to follow Semantic based and ontology for categorizing job vacancy announcements.

- There are job positions that state common educational background or requirement for the specific vacancy. For example, የመጀመሪያ ዲግሪ በግንባታ አስተዳደር፣ በአስተዳደር፣ በግብይት ወይም ተዛማጅ መስኮች which means "first degree in construction management, management, marketing or related fields" for "Construction Manager" position. The presence of these different fields of study in a single document can challenge the constructed categorization model and affect the performance of the classification model. Hence, we recommend to combine job titles and job qualification or educational background as input to minimize the problem.

- In this study, SVM, KNN and NB classification algorithms were employed for categorizing vacancy text. It is recommended to test other classification algorithms for categorizing vacancy text.

- There are little works done on stemming Amharic words. In order to overcome the stemming difficulty during text classification there is a need to conduct further study to design and develop a full-fledged automatic Amharic stemmer algorithm.

- There is also a need to design job vacancy text categorization for other local languages, such as Afaan Oromo, Tigrigna and other local languages in which such announcement may appear.

References

[1] S. Teklu, "Automatic categorization of Amharic news text: a machine learning approach," Addis Ababa University, Doctoral dissertation, Addis Ababa, Ethiopia, 2003.

[2] Zhang, Shilin, Heping Li, and Shuwu Zhang. , "Job opportunity finding by text classification," *Procedia Engineering,* vol. 29, no. 1, pp. 1528-1532, 2012.

[3] Gebrehiwot Assefa, Berhe, "A two step approach for Tigrigna text categorization," Unpublished Masters Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2011.

[4] A. Kumilachew, "Hierarchical Amharic News Text Classification," Unpublished MSc Thesis. Addis Ababa University, Addis Ababa, Ethiopia, 2010.

[5] A. Diriba, "Automatic Classification of AFAAN Oromo News Text: The Case of Radio Fana," Unpublished MSc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2009.

[6] Ikonomakis, M., Sotiris Kotsiantis, and V. Tampakas. ", "Text classification using machine learning techniques," *WSEAS transactions on computers,* vol. 4, no. 8 , pp. 966-974., 2005.

[7] Defersha, Naol Bakala, and Getachow Mamo. ", "A Two Steps Approach for Afan Oromo Nonfiction Text Categorization," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* vol. 3, no. 1, pp. 2456-3307, 2018.

[8] Slavazza, P., "What is the best method for automatic text classification," Medium, 05 September 2019. [Online]. Available: https://towardsdatascience.com. [Accessed 13 ebruary 2020].

[9] A. Ozgur, "Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization," Unpublished Master’s Thesis, Boğaziçi University , İstanbul, 2004.

[10] Blumberg, Robert, and Shaju Atre, "Automatic classification: moving to the mainstream," *DM REVIEW,* vol. 13, pp. 12-19, 2003.

[11] Ian, H. Witten, and Frank Eibe., "Data mining: Practical machine learning tools and technique," 2005.

[12] Grace, G. Hannah, and Kalyani Desikan, ""Experimental estimation of number of clusters based on cluster quality," *arXiv preprint arXiv,* vol. 1, no. 2, pp. 304-315, 2015.

[13] MonkeyLearn, "text-classifiers," MonkeyLearn, 15 May 2019. [Online]. Available: https://monkeylearn.com. [Accessed 12 February 2020].

[14] Javed, Faizan, Matt McNair, Ferosh Jacob, and Meng Zhao, "Towards a job title classification system," *arXiv preprint arXiv,* vol. 1606, no. 00917, 2016.

[15] Addis, Andrea, "Study and development of novel techniques for hierarchical text categorization," University of Cagliari, Sardinia, 2010.

[16] Sun, Aixin, and Ee-Peng Lim, "Hierarchical text classification and evaluation," in *Proceedings 2001 IEEE International Conference on Data Mining*, IEEE, 2001.

[17] A. B. ASRES, "A SEMI- SUPERVISED APPROACH FOR AMHARIC NEWS CLASSIFICATION," Unpublished Msc. Thesis, Addis Ababa University, Addis Ababa, 2012.

[18] F. e. a. Amato, "Classification of web job advertisements: A case study," in *CRISP Research Centre, Univerisity of Milan-Bicocca*, Curran, Italy, 2015.

[19] W. Kelemework, "Automatic Amharic text news classification: Aneural networks approach," *Ethiopian Journal of Science and Technology,* vol. 6, no. 2, pp. 127-137, 2013.

[20] Lynch, John, "An Analysis of Predicting Job Titles Using Job Descriptions," Technological University Dublin, MSc. Thesis, Dublin, 2017.

[21] Berndtsson, Mikael, Jörgen Hansson, Björn Olsson, and Björn Lundell, Thesis projects: a guide for students in computer science and information systems, Verlag London: Springer Science & Business Media, 2007.

[22] "Why Python," in *Hands-on Python Tutorial*, Loyola University Chicago, amazonaws.com, 2020, p. 4.

[23] K. Jain, "Scikit-learn(sklearn) in Python," Analytics Vidhya, 5 January 2015. [Online]. Available: https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/. [Accessed 10 08 2020].

[24] Hossin, Mohammad, and M. N. Sulaiman, " A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process,* vol. 5, no. 2, pp. 1-11, 2015.

[25] Ko, Youngjoong, and Jungyun Seo, "Automatic text categorization by unsupervised learning," in *COLING 2000 Volume 1*, The 18th International Conference on Computational Linguistics, 2000.

[26] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR),* vol. 1, no. 34, pp. 1-47, 2002.

[27] A. Faraz, "AComparison OF TEXT CATEGORIZATION METHODS," *International Journal on Natural Language Computing.-2016.-5 (1).-PP: 31-44.,* vol. 1, no. 5, pp. 31-44, 2016.

[28] A. Ozgur, "Supervised and unsupervised machine learning techniques for text document categorization," Unpublished Master's Thesis, İstanbul: Boğaziçi University, Istanbul, 2004.

[29] https://monkeylearn.com, "text-classification," MonkeyLaern, 15 02 2019. [Online]. Available: https://monkeylearn.com/text-classification/. [Accessed 01 07 2020].

[30] Liu, Bing, et al, "Text classification by labeling words," *AAA,* vol. 4 , pp. 425-430, 2004.

[31] Dalal, Mita K., and Mukesh A. Zaveri , "Automatic text classification: a technical review," *International Journal of Computer Applications,* vol. 2, no. 28, pp. 37-40., 2011.

[32] https://www.mathworks.com, "machine-learning.html," MathWorks, [Online]. Available: https://www.mathworks.com/discovery/machine-learning.html. [Accessed 15 07 2020].

[33] Kamal Mohammed Jimalo, Ramesh Babu, Yaregal Assabie, "Afaan Oromo News Text Categorization using Decision Tree Classifier and Support Vector achine: A Machine Learning Approach," *nternational Journal of Computer Trends and Technology (IJCTT),* vol. 47, no. 1, pp. 2231-2803, 2017.

[34] Sonoo Jaiswal, "unsupervised-machine-learning," Java T Point, 21 January 2019. [Online]. Available: https://www.javatpoint.com/unsupervised-machine-learning. [Accessed 12 May 2020].

[35] M. Allard, "Introduction to semi-supervised learning and adversarial training," Inside Machine Learning, 23 05 2019. [Online]. Available: https://medium.com/inside-machine-learning/placeholder-3557ebb3d470. [Accessed 20 07 2020].

[36] N. a. N. T. ". .. 2. Sloanim, "The Power of Word Clustering for Text Classification.," in *Proceedings of European Colloquium on IR Research*, ECIR, 2001.

[37] Popa IS, Zeitouni K, Gardarin G, Nakache D, Métais E. , "Text categorization for multi-label documents and many categories(pp. 421-426)," in *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*, Washington DC, USA: IEEE., 2007.

[38] Tripathy, Abinash, Agrawal A., and Rath S. K, "Classification of Sentimental Reviews Using Machine Learning Techniques," *Procedia Computer Science,* vol. 57, pp. 821-829, 2015.

[39] Peng, Fuchun, and Dale Schuurmans, "Combining naive Bayes and n-gram language models for text classification," in *European Conference on Information Retrieval, pp. 335-350.* , Springer, Berlin, Heidelberg, , 2003.

[40] S. Schrauwen, "Machine learning approaches to sentiment analysis using the dutch netlog corpus," in *Computational Linguistics and Psycholinguistics Research Center* , Antwerp, Belgium, 2010.

[41] Pratiwi, Oktariani Nurul, Budi Rahardjo, and Suhono Harso Supangkat, "Attribute Selection Based on Information Gain for Automatic Grouping Student System," in *International Conference on Soft Computing, Intelligence Systems, and Information Technology (pp. 205-211)*, Springer, Berlin, Heidelberg, 2015.

[42] S. Ray, "understaing-support-vector-machine-example-code," Analytics Vidhya, 13 09 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code. [Accessed 07 16 2020].

[43] Han, Jiawei, Micheline Kamber, and Jian Pei, "Data mining: concepts and techniques," pp. 350-375, 2006.

[44] García-Gonzalo, Esperanza, et al, "García-Gonzalo, Esperanza, Zulima Fernández-Muñiz, Paulino José García Nieto, Antonio Bernardo Sánchez, and Marta Menéndez Fernández," vol. 7, no. 9, p. 531, 2016.

[45] Dey L, Chakraborty S, Biswas A, Bose B, Tiwari S. , "Sentiment analysis of review datasets using naive bayes and k-nn classifier," *arXiv preprint arXiv:,* vol. 1610, p. 09982, 2016.

[46] Nikhath, A. et. al., "Building a K-Nearest Neighbor Classifier for Text Categorization," *International Journal of Computer Science and Information Technologies,* vol. 1, no. 7, pp. 254-256, 2016.

[47] Srivastava, Tavish, "How does Artificial Neural Network (ANN) algorithm work? Simplified!," Analytics Vidhya, 20 October 2014. [Online]. Available: https://www.analyticsvidhya.com/blog/2014/10/ann-work-simplified/. [Accessed 23 December 2020].

[48] P. e. a. Prasanna, "Text classification using artificial neural networks," *International Journal of Engineering & Technology,* vol. 7, no. 1, pp. 603-606, 2018.

[49] Tripathi, Kshitij, Rajendra G. Vyas, and Anil K. GuptaDocument Classification Using Artificial Neural Network, "Document Classification Using Artificial Neural Network," *Asian Journal of*

Computer Science and Technology, vol. 8, no. 2, pp. 55-58, 2019.

[50] upgrad, "neural-network-tutorial-step-by-step-guide-for-beginners," upGrad blog, 20 nov 2019. [Online]. Available: https://www.upgrad.com/blog/neural-network-tutorial-step-by-step-guide-for-beginners/. [Accessed 10 sep 2021].

[51] Jindal, Rajni, and Shweta Taneja, "A lexical approach for text categorization of medical documents," *Procedia Computer Science,* vol. 26, pp. 314-320, 2015.

[52] Suleymanov, Umid, and Samir Rustamov, "Automated News Categorization using Machine Learning methods," in *IOP Conf. Ser. Mater. Sci. Eng.*, Baku, Azerbaijan , 2018.

[53] Atelach Alemu, "Automatic Sentence Parsing for AmharicText an Experiment using probabilistic Context Free Grammars.," Addis Ababa University, Masters Thesis,, Addis Ababa, 2002.

[54] Bender, Marvin Lionel, "Language in Ethiopia," in *Oxford University Press*, London, 1976.

[55] A. IBRAHIM, "A HYBRID APPROACH TO AMHARIC BASE PHRASE CHUNKING AND PARSING," Unpublished MSc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2013.

[56] T. Tensou, "Word Sequence Prediction for Amharic Language," Unpublished MSc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2014.

[57] Abate, Mesfin, and Yaregal Assabie, "Development of Amharic morphological analyzer using memory-based learning," in *International Conference on Natural Language Processing. Springer*, Cham, 2014.

[58] W. a. M. G. Mulugeta, "Learning morphological rules for Amharic verbs using inductive logic programming," *Language Technology for Normalisation of Less-Resourced Languages,* vol. 7, 2012.

[59] A. Gonfalonieri, "How to Build A Data Set For Your Machine Learning Project," Medium, 14 Feb 2019 . [Online]. Available: https://towardsdatascience.com/how-to-build-a-data-set-for-your-machine-learning-project-5b3b871881ac. [Accessed 08 08 2020].

[60] cloudfactory, "The Essential Guide to Quality Training Data for Machine Learning," cloudfactory, [Online]. Available: https://www.cloudfactory.com/training-data-guide. [Accessed 14 08 2020].

[61] C. W. Store, "Data Scraper - Easy Web Scraping," Chrome Web Store, 8 March 2019. [Online]. Available: https://chrome.google.com/webstore/detail/data-scraper-easy-web-scr/nndknepjnldbdbepjfgmncbggmopgden. [Accessed 19 07 2020].

[62] DataMiner, "How Data Miner Works," DataMiner, [Online]. Available: https://data-miner.io/how-it-works. [Accessed 19 07 2020].

[63] Haddi, et al., "The role of text pre-processing in sentiment analysis," *Procedia Computer Science,* vol. 17, pp. 26-32, 2013.

[64] codecademy, "Text Preprocessing," codecademy, 2020. [Online]. Available: https://www.codecademy.com/learn/natural-language-processing/modules/nlp-text-preprocessing?utm_source=rakuten&utm_medium=affiliate&utm_campaign=adgoal.net&utm_content=10-1&ranMID=44188&ranEAID=a1LgFw09t88&ranSiteID=a1LgFw09t88-TtCWE7dB.4Cz1CPnzPF1VQ. [Accessed 1 September 2020].

[65] Data Monsters, "Text Preprocessing in Python: Steps, Tools, and Examples," Medium, 16 October 2018. [Online]. Available: https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908. [Accessed 1 September 2020].

[66] A. Navlani, "Text Analytics for Beginners using NLTK," Tutorials, 13 December 2019. [Online]. Available: https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk. [Accessed 31 August 2020].

[67] J. Lynch, "An Analysis of Predicting Job Titles Using Job Descriptions," Technological University Dublin, Ireland, 2017.

[68] S. Singh, "NLP Essentials:Removing Stopwords and Performing Text Normalization using NLTK and spaCy in Python," Analytics Vidhya, 21 August 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/08/how-to-remove-stopwords-text-normalization-nltk-spacy-gensim-python/. [Accessed 31 August 2020].

[69] S. Jain, "Introduction to Stemming," GeeksforGeeks, 17 September 2020. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-stemming/. [Accessed 17 September 20].

[70] A. Navlani, "naive-bayes-scikit-learn," www.datacamp.com, 4 december 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn. [Accessed 4 june 2021].

[71] M. McGregor, "SVM Machine Learning Tutorial," #Machine learning, 1 July 2020. [Online]. Available: https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/. [Accessed 17 September 2020].

[72] J. T. Point, "K-Nearest Neighbor(KNN) Algorithm for Machine Learning," Java T Point, 06 January 2019. [Online]. Available: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning. [Accessed 23 September 2020].

[73] R. Agarwal, "The 5 Classification Evaluation metrics every Data Scientist must know," Medium, 17 Sep 2019 . [Online]. Available: https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226. [Accessed 17 September 2020].

[74] A. Kumilachew, "Hierarchical Amharic News Text Classification," Unpublished MSc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2010.

[75] Y. Zhao, "Comparison of Agglomerative and Partitioning Document Clustering Algorithms," in *ACM Press*, Washington DC, 2002.

[76] T. S. Madhulatha, "An overview on clustering method," *arXiv preprint arXiv,* vol. 1205, no. 1117 , 2012.

[77] Rokach, Lior, and Oded Maimon, "Clustering Methods," in *Data mining and knowledge discovery handbook*, Tel Aviv, Israel, Springer, Boston, MA, 2005, pp. 321-352.

[78] Abualigah, Laith Mohammad, Ahamad Tajudin Khader, and Essam Said Hanandeh, "A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis," *Engineering Applications of Artificial Intelligence,* vol. 73, pp. 111-125, 2018.

[79] D. Kalita, "Supervised and Unsupervised Document Classification: A Survey," *International Journal of Computer Science and Information Technologies,* vol. 6, no. 2, pp. 1971-1974, 2015.

[80] Dharmarajan, A., and T. Velmurugan, "Efficiency of k-Means and k-Medoids Clustering Algorithms using Lung Cancer Dataset," *International Journal of Data Mining Techniques and Applications,* vol. 5, no. 2, pp. 150-156, 2016.

[81] L. a. J. Q. Xu, ""Unsupervised multi-class sentiment classification approach," *O KNOWLEDGE ORGANIZATION,* vol. 46, no. 1, pp. 15-32, 2019.

[82] A. Soni, "Clustering with Gaussian Mixture Model," Medium, 05 December 2017. [Online]. Available: https://medium.com/clustering-with-gaussian-mixture-model/clustering-with-gaussian-mixture-model-c695b6cd60da. [Accessed 23 07 2020].

[83] E. a. C. R. Acuna, ""The treatment of missing values and its effect on classifier accuracy Classification, clustering, and data mining applications," *Springer, Berlin, Heidelberg,* pp. 639 -647, 2004.

[84] P. Grg, "Getting started with NLP using NLTK," Medium, 6 September 2018. [Online]. Available: https://becominghuman.ai/nlp-for-beginners-using-nltk-f58ec22005cd. [Accessed 01 September 2020].

[85] J. Gallagher, "Python Lowercase: A Step-By-Step Guide," KAREER KARMA, 3 March 2020. [Online]. Available: https://careerkarma.com/blog/python-lowercase. [Accessed 1 September 2020].

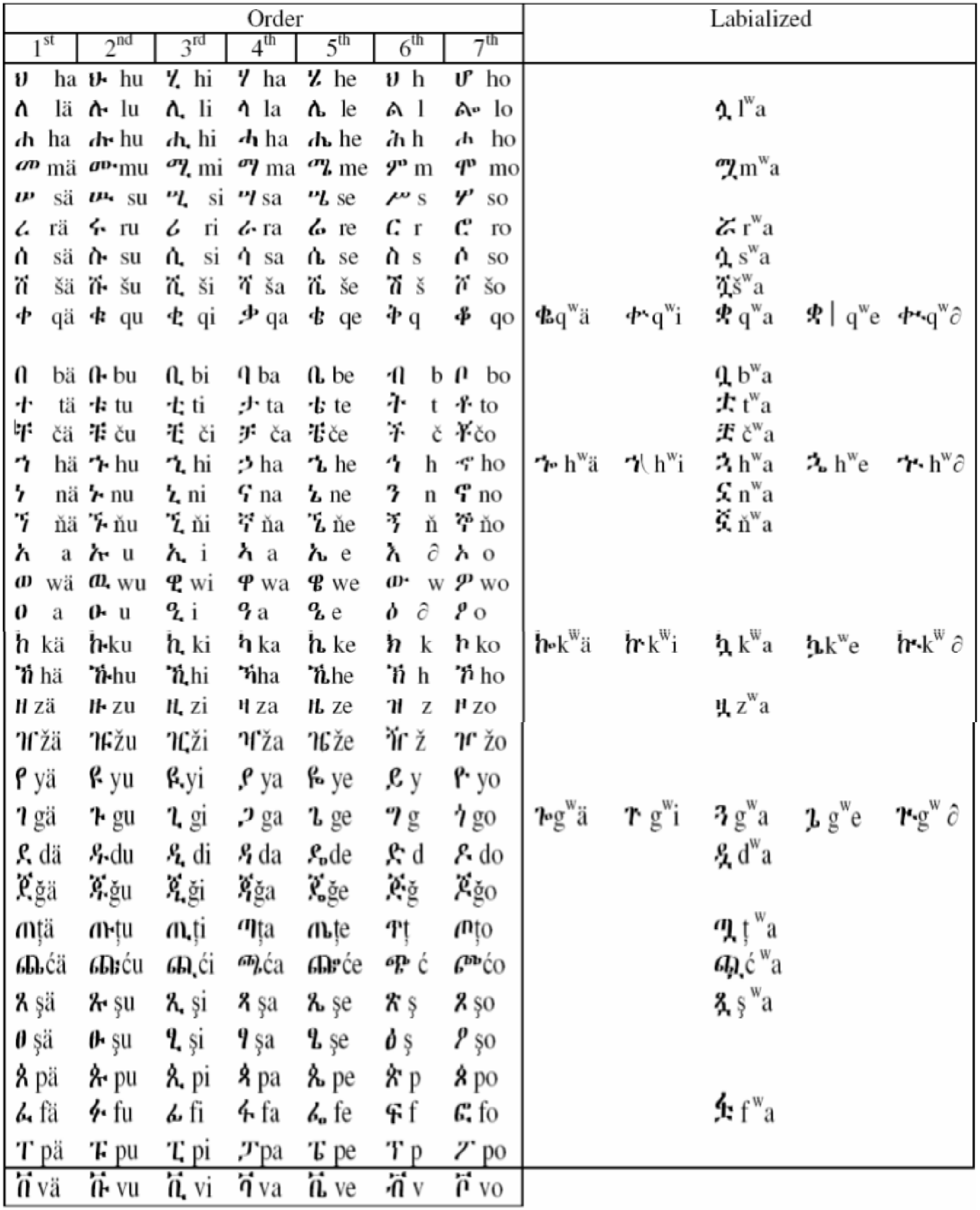
[86] M. Mayo, "A General Approach to Preprocessing Text Data," KDnuggets, 12 December 2017. [Online]. Available: https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html. [Accessed 31 August 2020].

[87] M. Kindeneh, "Opinion Mining for Amhara Broadcasting Agency News," Unpublished Masters Thesis, Bahir Da University, Ethiopia, Bahir Dar , Ethiopia, 2020.

[88] A., Nigam, et al., "Text Classification from Labeled and Unlabeled Documents Using EM," *Kluwer Academic Publishers,* vol. 39, no. 2, p. 103–134, 2000.

Appendix

Appendix I: Amharic characters ('Fidel') (adapted: from [4])

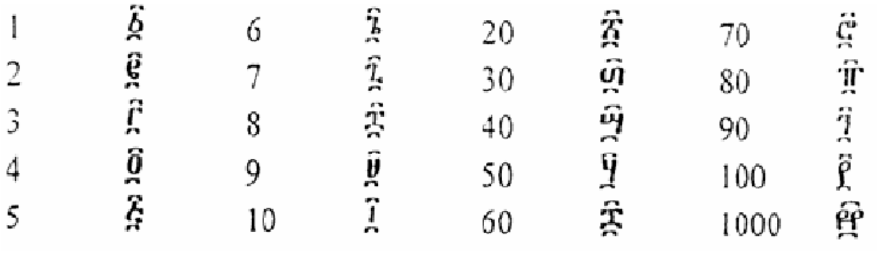


Order							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ ha	ሁ hu	ሂ hi	ሃ ha	ሄ he	ህ h	ሆ ho					
ለ lä	ሉ lu	ሊ li	ላ la	ሌ le	ል l	ሎ lo			ሏ l^{w}a		
ሐ ha	ሑ hu	ሒ hi	ሓ ha	ሔ he	ሕ h	ሖ ho					
መ mä	ሙ mu	ሚ mi	ማ ma	ሜ me	ም m	ሞ mo			ሟ m^{w}a		
ሠ sä	ሡ su	ሢ si	ሣ sa	ሤ se	ሥ s	ሦ so					
ረ rä	ሩ ru	ሪ ri	ራ ra	ሬ re	ር r	ሮ ro			ሯ r^{w}a		
ሰ sä	ሱ su	ሲ si	ሳ sa	ሴ se	ስ s	ሶ so			ሷ s^{w}a		
ሸ šä	ሹ šu	ሺ ši	ሻ ša	ሼ še	ሽ š	ሾ šo			ሿ š^{w}a		
ቀ qä	ቁ qu	ቂ qi	ቃ qa	ቄ qe	ቅ q	ቆ qo	ቈ q^{w}ä	ቊ q^{w}i	ቋ q^{w}a	ቌ q^{w}e	ቍ q^{w}∂
በ bä	ቡ bu	ቢ bi	ባ ba	ቤ be	ብ b	ቦ bo			ቧ b^{w}a		
ተ tä	ቱ tu	ቲ ti	ታ ta	ቴ te	ት t	ቶ to			ቷ t^{w}a		
ቸ čä	ቹ ču	ቺ či	ቻ ča	ቼ če	ች č	ቾ čo			ቿ č^{w}a		
ኀ hä	ኁ hu	ኂ hi	ኃ ha	ኄ he	ኅ h	ኆ ho	ኈ h^{w}ä	ኊ h^{w}i	ኋ h^{w}a	ኌ h^{w}e	ኍ h^{w}∂
ነ nä	ኑ nu	ኒ ni	ና na	ኔ ne	ን n	ኖ no			ኗ n^{w}a		
ኘ ňä	ኙ ňu	ኚ ňi	ኛ ňa	ኜ ňe	ኝ ň	ኞ ňo			ኟ ň^{w}a		
አ a	ኡ u	ኢ i	ኣ a	ኤ e	እ ∂	ኦ o					
ወ wä	ዉ wu	ዊ wi	ዋ wa	ዌ we	ው w	ዎ wo					
ዐ a	ዑ u	ዒ i	ዓ a	ዔ e	ዕ ∂	ዖ o					
ከ kä	ኩ ku	ኪ ki	ካ ka	ኬ ke	ክ k	ኮ ko	ኰ k^{w}ä	ኲ k^{w}i	ኳ k^{w}a	ኴ k^{w}e	ኵ k^{w}∂
ኸ hä	ኹ hu	ኺ hi	ኻ ha	ኼ he	ኽ h	ኾ ho					
ዘ zä	ዙ zu	ዚ zi	ዛ za	ዜ ze	ዝ z	ዞ zo			ዟ z^{w}a		
ዠ žä	ዡ žu	ዢ ži	ዣ ža	ዤ že	ዥ ž	ዦ žo					
የ yä	ዩ yu	ዪ yi	ያ ya	ዬ ye	ይ y	ዮ yo					
ገ gä	ጉ gu	ጊ gi	ጋ ga	ጌ ge	ግ g	ጎ go	ጐ g^{w}ä	ጒ g^{w}i	ጓ g^{w}a	ጔ g^{w}e	ጕ g^{w}∂
ደ dä	ዱ du	ዲ di	ዳ da	ዴ de	ድ d	ዶ do			ዷ d^{w}a		
ጀ ǧä	ጁ ǧu	ጂ ǧi	ጃ ǧa	ጄ ǧe	ጅ ǧ	ጆ ǧo					
ጠ ţä	ጡ ţu	ጢ ţi	ጣ ţa	ጤ ţe	ጥ ţ	ጦ ţo			ጧ ţ ^{w}a		
ጨ ćä	ጩ ću	ጪ ći	ጫ ća	ጬ će	ጭ ć	ጮ ćo			ጯ ć ^{w}a		
ጸ şä	ጹ şu	ጺ şi	ጻ şa	ጼ şe	ጽ ş	ጾ şo			ጿ ş ^{w}a		
ፀ şä	ፁ şu	ፂ şi	ፃ şa	ፄ şe	ፅ ş	ፆ şo					
ጰ pä	ጱ pu	ጲ pi	ጳ pa	ጴ pe	ጵ p	ጶ po					
ፈ fä	ፉ fu	ፊ fi	ፋ fa	ፌ fe	ፍ f	ፎ fo			ፏ f^{w}a		
ፐ pä	ፑ pu	ፒ pi	ፓ pa	ፔ pe	ፕ p	ፖ po					
ቨ vä	ቩ vu	ቪ vi	ቫ va	ቬ ve	ቭ v	ቮ vo					

Appendix II: Amharic punctuation marks (Adapted from: [4])

No.	Punctuation mark	Symbol	Purpose
1	The four dots or double colon	::	Mark end of a sentence
2	Colon	:	Separate words in a sentence: not common
3	White space		Separate words in a sentence: current practice
4	Question mark	?	Placed at the end of questions
5	Exclamation mark	!	Used at the end of sentences that show exclamation
6	Comma	፣	Used like comma
7	Semi-colon	፤	Used like semi-column
8	Three dots	…	For deliberate omission of words, phrases, or sentences
9	Quotation marks	‹‹ ››	Used at the beginning and at the end of quoted word, phrase, etc.
10	Parenthesis	( )	To enclose elaboration
11	Stroke	/	Separate date, month, etc.
12	Mocking mark	⁝	Placed at the end of mocking sentence

Appendix III: Amharic Numbers (adapted from: [4])



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

Appendix IV: Sample Python Code

```
#!/usr/bin/env python
# coding: utf-8
import pandas as pd
from sklearn import model_selection, naive_bayes, svm
from sklearn.metrics import accuracy_score
from matplotlib import pyplot as plt
from sklearn.metrics import classification_report
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
df=pd.read_excel("file_location ")
#df=pd.read_csv("file_location")
df.head()
for index,entry in enumerate(df['requirement']):
    df.loc[index,''] = str(df)
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
#plt.scatter(df['category'], df['requirement'])
x=df['requirement']
y=df['category']
#print (x)
from sklearn.model_selection import train_test_split
X_train,X_test, Y_train, Y_test=train_test_split(x, y, test_size=0.2, random_state=5)
Tfidf_vect = TfidfVectorizer(max_features=5000)
Tfidf_vect.fit(df[''])
```

```
Train_X_Tfidf = Tfidf_vect.transform(X_train)

Test_X_Tfidf = Tfidf_vect.transform(X_test)

# fit the training dataset on the classifier

KNN= KNeighborsClassifier()

KNN.fit(Train_X_Tfidf,Y_train)

predictions_KNN = KNN.predict(Test_X_Tfidf)

print("KNN Accuracy Score -> ",accuracy_score(predictions_KNN, Y_test)*100)

print(classification_report(Y_test, predictions_KNN))
```