Signals and Communication Technology

Virender Kadyan
Amitoj Singh
Mohit Mittal
Laith Abualigah *Editors*

Deep Learning Approaches for Spoken and Natural Language Processing

Springer

Signals and Communication Technology

This series is devoted to fundamentals and applications of modern methods of signal processing and cutting-edge communication technologies. The main topics are information and signal theory, acoustical signal processing, image processing and multimedia systems, mobile and wireless communications, and computer and communication networks. Volumes in the series address researchers in academia and industrial R&D departments. The series is application-oriented. The level of presentation of each individual volume, however, depends on the subject and can range from practical to scientific.

Indexing: All books in "Signals and Communication Technology" are indexed by Scopus and zbMATH

For general information about this book series, comments or suggestions, please contact Mary James at mary.james@springer.com or Ramesh Nath Premnath at ramesh.premnath@springer.com.

More information about this series at http://www.springer.com/series/4748

Virender Kadyan • Amitoj Singh • Mohit Mittal
Laith Abualigah
Editors

Deep Learning Approaches for Spoken and Natural Language Processing

Springer

Editors
Virender Kadyan
University of Petroleum and Energy Studies
Dehradun, India

Amitoj Singh
Jagat Guru Nanak Dev Punjab State Open University
Patiala, Punjab, India

Mohit Mittal
Kyoto Sangyo University
Kyoto, Japan

Laith Abualigah
Amman Arab University
Amman, Jordan

ISSN 1860-4862 ISSN 1860-4870 (electronic)
Signals and Communication Technology
ISBN 978-3-030-79777-5 ISBN 978-3-030-79778-2 (eBook)
https://doi.org/10.1007/978-3-030-79778-2

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To God, my teachers, and my family

Acknowledgment

We would like to thank the contributors who significantly contributed to this book. I also want to thank my co-editors; they clarified my initially vague concepts and assisted in a number of processes including reviewing, copyediting, diagramming, etc. In addition, I would like to express my deep appreciation and sincere gratitude to the Mr. Shabib Shaikh Project Coordinator (Books) and Mary E. James Senior Editor for their tireless efforts in supporting our ideas. I would like to extend my thanks to the entire team, and particularly to Dr. Virender Kadyan for their enthusiasm and cooperation. Without the guidance of my co-editors Dr. Mohit Mittal and Dr. Laith Abualigah, this project would not have been possible.

Contents

Abbreviations

AI	Artificial Intelligence
ANN	Artificial neural network
ASR	Automatic speech recognition
BFCC	Basilar-membrane frequency-band cepstral coefficient
BNs	Bayesian networks
CBOW	Continuous bag of words
CBOW	Continuous bag of words model
CER	Character error rate
CNN	Convolution neural network
DCT	Discrete cosine transform
DFD	Data flow diagrams
DFT	Discrete Fourier transform
DNF	Disjunctive normal form
DNN	Deep neural networks
DPIL	Detecting paraphrases for Indian languages
ELM	Extreme learning machine
FD	Fractal dimension
FFT	Fast Fourier transform
FSK	Frequency shift keying
GA	Genetic algorithm
GFCC	Gammatone frequency cepstral coefficients
GMM-UBM	Gaussian Mixture Model-Universal Background Model
GPU	Graphical Processing Unit
HBD	Hausdorff–Besicovitch dimension
HLDA	Heteroscedastic linear discriminant analysis
HMM	Hidden Markov models
IDE	Integrated development environments
IEA	Intelligent essay assessor
LD	Levenshtein distance
LPCC	Linear prediction coding coefficient

L-RNN	Layered recurrent neural networks
LSA	Latent semantic analysis
MAE	Mean absolute error
MFCC	Mel frequency cepstral coefficients
ML	Machine learning
MT	Machine translation
MTL	Multi-task learning
NB	Naïve Bayes
NL	Natural language
NLP	Natural language processing
NN	Neural networks
PCA	Principle components analysis
PCR	Pearson correlation results
PLP	Perceptual linear prediction
PNN	Probabilistic neural networks
POS	Part-of-speech
PSO	Particle swarm optimization
RAE	Recursive auto encoders
RNN	Recurrent neural network
SA	Sentiment analysis
SRS	Software requirements specification
STFT	Short-term Fourier transformation
SVM	Support vector machine
TF	Binary rerm frequency
TF	Term frequency
TF-IDF	Term frequency-inverse document frequency
TPU	Tensor Processing Unit
UML	Unified modeling language
WCM	Word-by-context matrix

Survey on Twitter Sentiment Analysis: Architecture, Classifications, and Challenges

Laith Abualigah, Nada Khaleel Kareem, Mahmoud Omari, Mohamed Abd Elaziz, and Amir H. Gandomi

1 Introduction

Sentiment analysis (SA) is that area for studying knowledge or interpreting people's opinions toward a particular topic [1]. It is also called opinion mining because it interprets the speaker's opinion on a specific topic [2]. It has various names and include different tasks; among those are affect analysis, opinion mining, sentiment mining, subjectivity analysis, and others. Each name has its own job and its diverse tasks, but they all meet to obtain the feelings and opinions of people on a specific topic [1]. In other words, it determines whether the opinion toward a specific topic is negative, positive, or neutral. Hence, sentiments are classified into three categories: negative, positive, and neutral sentiments [2]. Positive sentiments are the good terms about the topic in consideration. When the positive impressions are high, it concludes good feelings. Negative sentiments, on the other hand, are the bad terms about the topic in consideration. For example, many business owners use Twitter to track and monitor people's opinions about their products and services. When positive feedbacks about a product are high, then the expected purchase rate would be high. On the other hand, when negative impressions are high, it is rejected from the

L. Abualigah (✉)
Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia

N. K. Kareem · M. Omari
Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

M. A. Elaziz
Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt

A. H. Gandomi
Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_1

preference list, and no purchases are expected for the product. Finally, neutral sentiments are neither good nor bad terms about the topic. Hence, it is neither favor nor neglected.

SA is a natural language processing (NLP) task that extracts natural data in which it focuses on obtaining feelings. In general, it focuses on inferring from the behavior of the speaker or person regarding a particular topic [3]. The field of sentiment analysis is multidisciplinary, as it deals with individuals' feelings, opinions, emotions, and attitude toward products, services, topics, issues, individual, and anything else subject to opinion. Subjects of sentiment analysis include various areas like computer languages, NLP, machine learning (ML), and technical intelligence as well as information retrieval. It includes a set of computational and natural languages based on techniques that can be used to extract data from a specific text diverse to subjective opinions or feelings [4]. The field of SA is a subdomain of ML. With manual training, the problem can be solved to analyze feelings to the required level as there is no automated integrated system for analyzing feelings without requiring manual intervention [5].

Different levels can be applied to sentiment analysis, including the *document level* that gives one polarity to the entire document, the *sentence level* that gives polarity to each sentence, and the *entity/aspect level* that is based on analyzing each word that has feelings. Previously, sentiment analysis was limited to knowing the polarity of opinions. It was of no use in making or taking decisions, as no reasons were known about why the sentiments have changed. Hence, there is a need to build systems to explain these differences in public sentiments. Several studies of multiple sentimental techniques and various algorithms are used to analyze feelings [2, 3]. The main resource for sentiment analysis is web data, which is huge and the largest store of unstructured and structured information.

Nowadays, social networks have become widely used. Facebook, Twitter, LinkedIn, and YouTube in addition to other sites are very popular. Twitter is the most used platform in social media, which is a microblog that permits its users to post their feelings and opinions. The number of Twitter registrants in 2017 reached nearly 696 million, and the amount of tweets per day reached approximately 58 million tweets. Such microblogging has become an essential and important source of great value to people's opinions and feelings. It includes various topics and different directions, including political and economic in addition to religious and social as well as sports and other trends. Such important data can be used efficiently in studying market conditions, social studies, disease surveillance, and other common topics. Twitter users are not only, regular users but also leaders heads of state, firm's executives, and celebrities [1, 6]. For example, if you want to know the people's opinion about the former US President Barack Obama, you can refer to social network sites such as Twitter. The Twitter platform includes millions of opinions about what Obama has done during his presidency. You would find positive, negative, and neutral opinion tweets. Accuracy can be obtained in the answer to whether the people believe that he fulfilled his duties or not by extracting some accurate words from the tweets that indicate the opinion on this matter [2].

Obviously, it is necessary to collect and analyze the data represented by text posts on various matters in order to reach the feelings expressed in the tweets. Most of the data available in networks is irregular, accounting for approximately 80% of all data in the world. It is difficult to obtain accurate information, make a judgment, or analyze this data. Sentiment analysis is of great importance in extracting or mining opinions, as it helps to reveal people's feelings or opinions from social media, which is used for sharing opinions and ideas between people linked on the global web [1, 6]. The importance of this study stems from the importance of SA to all people's endeavor [2]. This study will describe the process of SA and its dependency on various levels of text analysis, namely, sentence level, document level, and phrase level. We will also look at the architecture of the sentiment analysis process, which includes four stages: data collection, preprocessing, feature extraction, and training and classification. We will also discuss more broadly the most important algorithms used in this field. The challenges that face sentiment analysis will be described. Finally, we will remark some gaps that assist to expand the scope of progression in this research area.

This chapter is organized as follows. Section 2 describes the different levels of sentiment analysis. Section 3 describes the architecture used for performing a comprehensive and detailed sentiment analysis. The various algorithms used in this field are discussed in Sect. 4. In Sect. 5, we will discuss some of the challenges facing SA. Section 6 concludes the study and gives further research direction.

2 Levels of Sentiment Analysis

Various studies on sentiment analysis have been done at three main levels [1–4]:

(a) *Document level analysis:* In this type [7], the entire document is viewed, and the goal is to analyze the overall feel of the document as the whole document is viewed as one subject. For example, when reviewing a specific product, the task will be to determine whether opinions are negative, positive, or neutral for that product. Sometimes, this level does not give correct results, as is the case in forums or blogs where one product is compared to another for the same characteristics. In addition, it is possible that the document contains sentences that are not related to the topic and should not be included in the process of analyzing feelings.

(b) *Phrase level analysis:* This level is limited to analyzing a complete sentence and decides whether it has positive, negative, or neutral feelings. A sentence may have two or more words. It is close to subjective classification; it carries accurate data from sentences that include the opinions and objective aspect. It may give inaccurate results if the sentence includes a negative sentence.

(c) *Entity/aspect level analysis:* It is also called the feature level. Instead of analyzing a document or sentence, this level is based on analyzing each word that has feelings. This level provides an accurate analysis of each aspect (feature level). It

analyzes the word to obtain different opinions. The word may be an adjective or an adverb and may be a noun. It depends on a concept; the opinion can be an attitude, a word, or a point of view.

3 System Architecture

Sentiment analysis is based on words that people use to express their opinion in a specific matter (negative, positive, or neutral). To reveal the direction of view, we first identify the meaningful words in the tweets and then discover their direction whether that word reflects positive feelings, negative feelings, or neutral feelings [8]. The system comprises four key stages: data collection, data processing, and feature extraction [1], as explained in Fig. 1. Data are obtained or collected from Twitter and preprocessed, which includes filtering to filter unique Twitter features and extracting side-based features to identify explicit and implicit aspects [9]. As an input, the system collects the tweet identifiers and the N-Gram model for the purpose of learning a class [10]. The tweets are normalized and converted to mono grams (four grams, three grams, two grams, and monograms) [11]. After preparing training and testing data, the different classifiers are applied to analyze the performance of the workbooks [6] and thus obtain the outputs. In the following, we discuss each stage in detail.

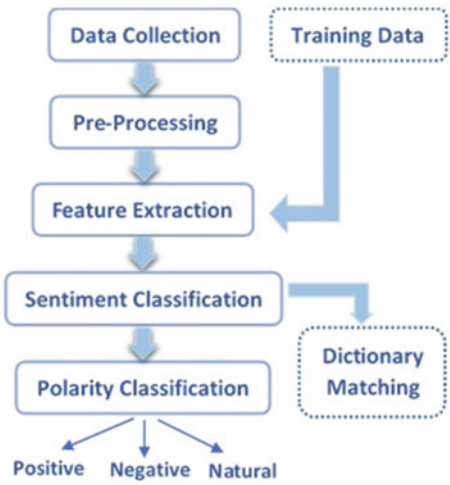

Fig. 1 Sentiment analysis architecture

3.1 Data Collection

It is important to collect data for a specific topic from relevant sites (as in microblogging) using queries [6]. This means that if we need to analyze feelings of opinion in the health field, for example, it is necessary to rely on sites specific to this field [12]. Blogs are sites that enable users to post messages, pictures, links to other sites, or videos. Messages posted on blogs are short, unlike traditional blogging. Currently, a number of platforms are available for blogs, including Twitter, LinkedIn, Google, Foursquare, and Tumblr [13]. Here, we will talk about Twitter as a source of data.

3.1.1 Twitter

Twitter, one of the most popular microblogging sites, is a form of platform and microblogging that permits its users to post messages denoted as tweets. Tweets contain numerous unique features [8, 14]. Twitter began in 2006 and has since attracted a large number of users. Ease of accessing and downloading publications and the amount of data it contains. Twitter was considered one of the largest datasets [13] that could be adopted in sentiment analysis. The adoption of Twitter data in the analysis of feelings is to classify the tweets into different feelings categories accurately [1]. Twitter data was adopted in various fields, as it was used as a source to monitor real-world outcomes or forecast information, including the analysis of extreme events such as Syria 2013, expected box office revenue for films, earthquakes in Japan, and others [12]. Twitter is characterized by some specific features that are listed below:

- *Tweet:* Tweet is the message that was posted on Twitter. The maximum message limit is approximately 140 characters. Tweet content (tweet topics) can differ from a personal opinion on a specific topic or news and may be in the form of links, news, and photos or may include videos.
- *Writing technology:* Because messages are short, many use abbreviations in comments, and some use symbols that give a lot of meaning. This is in addition to spelling errors, incorrect spelling in many tweets, and use of colloquial language.
- *Availability:* The number of people tweeting in the public domain on Twitter is large compared to other platforms such as Facebook (Facebook has many privacy settings). This made data widely available, as it is easy to collect tweets for training.
- *User/username*: When registering in the system, a name is chosen, and the name can be a pseudonym. This name is used in the system to post tweets.
- *Mention:* The Mention is when the tweet is being referred to another user, to share the topic with that user. The tweet uses the "@" symbol before the username that is indicated (@username).

- *Comments:* Comments often create conversation, which is the result of answering a comment, and other people are referred to.
- *Follower:* Followers are the ones who follow the user and his activities. Follow-up is the way to communicate with other Twitter users. Where the user receives an update from the followers and also sends his updates to the followers.
- *Retweet:* When a tweet is posted, another user can re-publish that tweet using retweet. It is considered a strength for disseminating information. It can be seen that the tweet was reposted with the abbreviation RT followed by a username (RTusername).
- *Hashtag:* This feature is used to classify the tweet and its relevance to a specific topic. The symbol # is followed by the name of the topic (#topic). The Hashtag is used by the tweet, thus access to all tweets using the same hashtag. Classification Hashtag are often popular topics.
- *Privacy:* This feature determines whether the tweet will be visible to everyone or only for followers. All of these characteristics that are mentioned are problems; on the other hand, these problems need to be processed, which we will address in the second stage [1, 8, 13, 14].

3.2 Preprocessing

Twitter tweets were adopted because they contain many opinions, which are presented in various ways. It has been categorized into positive and negative as well as neutral, which makes data analysis not difficult [15]. The representation of tweets is often in vague and informal ways [11]. And because of the diversity of language usage in tweets, it is possible that there are language or spelling errors. Tweets can include some symbols, abbreviations, usernames, links, and others that are not related to the classification process [9]. Therefore, processing techniques are used to obtain relevant content, while the rest of the comments away from the topic are ignored. This stage is very important in the classification process [15]. Hence, data quality has a significant impact on the results. So to enhance the analysis, the preliminary data are processed [15]. Among the most important processing steps that are implemented are as follows:

- *Tokenization:* After the tweets are compiled with identifiers available in the data sets, each tweet is broken down into a set of individual words. For each tweet, there will be a list of its own individual words [16–18].
- *Removal of non-English tweets:* The nature of the Twitter allows the use of more than 60 languages. The focus will be on the English language. We will remove non-English words and tweets.
- *Replace emoticons in many microblogging posts:* Many Twitter users use emoticons and shortcuts for tweets. Each of these symbols has strong connotations and is an indicator of feelings, as they are a concise way of identifying feelings. It will therefore have a vital part in determining the feelings of the tweet. It will be

Table 1 Some emoticons and their meaning

Emoticon	Meaning	Sentiment Class
:-D	Laughing	Positive
:-)	smile	Positive
o:)-	innocent	Positive
8-)	cool	Positive
:$	Happy blush	Positive
:(	defeated	Negative
:'(	Crying	Negative
:o	shocked	Negative
>(	Grumpy	Negative
(@)	Angry red	Negative
X|	Dead	Negative

easy to distinguish the polarity of messages, whether positive or negative. This is done using the emoticon dictionary, where the symbol is replaced with corresponding emotions (Table 1).

- *Removal of links/URL:* Due to the limited length of the text for tweets, users use URLs. These URLs do not carry any meaningful indications, as a word, within the tweet itself. But it does provide a large content of emotion that the user tries to express in a concise manner. However, it remains very difficult to reach the content of the URLs; therefore, the URL will be deleted.
- *Removal of target mentions:* Most of the time the user mentions another user in the tweet. It can be distinguished by the "@" symbol. It is placed in front of the username you want to refer to (@John). This part of the tweet (@John) is not important in the analysis because it does not have any moral significance. So it will be removed.
- *Removal of punctuations from hashtags:* The hashtags are important, providing a summary of what the tweet means. So to get the information, you must delete both punctuation and the symbol indicating it to retain only the important information from it.
- Removal of numbers: Sometimes, numbers are used in tweets. The numbers have no value when measuring feelings. Therefore, the numbers are removed from the tweet content.
- Handle sequences of repeated characters: Spelling correction is of great importance in analyzing feelings for tweet content. Often users express their opinions abnormally and loudly, without focusing on the correct texture and spelling. Tweeters use words like "cooooooool" or "wooooooow." In order to get the correct expression for such words, we replace the repeated letter more than three times with three letters to be "coool" and "wooow." We substitute three letters to distinguish between emphasized usage of the word and regular usage of the word; for example, the word "cool" gives another meaning. WordNet is used to ensure that unnecessary characters are removed.

- *Removal of stop words:* The tweets include many words that have no meaning, called stop words. Stop words do not contain any information about feelings and therefore are useless. When making tokenization, these words should be removed. Examples of stop words are a, an, the, and other words.
- Handle negative mentions: Negativity has an important and significant role in determining the tweet. The words "no," "not," "never," and others or the words that end with it should be replaced by a word referring to negation.
- *Uppercase identification:* It is common to use capital letters to express strong emotions. Such a type is called an e-shouting. It is a good indicator to get the message polarity easily. This step mines this feature before taking out casing [10, 11, 19, 20].

Note that we will need resources to process Twitter data including an emoticon dictionary and an acronym dictionary [21]. The dictionary of emoticon is for emoticons, which includes a number of the most used emoticons. As for the dictionary of acronym that is compiled from various sources and which includes translations of a large number of abbreviations, we will mention them in another section of this work.

3.3 Feature Extraction

The identification and selection of features are very significant for the classification of texts. We try to understand which features are the most significant for the classification process. Text feature extraction is the process of obtaining a list of words from previously processed data and then converting that list into a set of features that can be used by the classifier. A variety of methods for defining and extracting features from textual data are important for classification; some of these features are as follows [4, 19, 22, 23]:

- *Part-of-speech tagging (PoS)*: Important signals of opinion to find an adjective or a descriptive word for each sentence. The natural language processing technique uses pointers to the parts of speech. PoS indicator is an undertaking for the labeling of all words in a sentence to a PoS tag. These words relate to communal categories of English grammar including adjectives, verbs, names, and prepositions, as well as conjugation, pronouns, and interference. Because parts of speech define expressions of feelings and semantic relationships between expressions, they are used to filter features that indicate the direction of feelings.
- *N-gram model*: A set of texts related to a subject is analyzed. The texts represent tweets. Each word, or symbol, is extracted from tweets as a series of words, which is represented as N. At the end, a dictionary of words or symbols is formed. That sequence of symbols or words can be a letter, word, or byte and can be continuous symbols. Depending on the number of grams, the words are defined, which means 1-gram, which is also named a unigram, is composed of one symbol; 2-gram or bigram consists of two symbols, and trigram consists of three symbols. This

makes the analysis process capable of revealing the correlation between those words and the prominence of the phrase itself. For instance, the text "Microsoft is launching a new product" is composed of the resulting 2-gram word features: "Microsoft is," "is launching," "launching a," "a new", and "new product." The tweet is represented in N-gram, as in the previous example. These features are N-gram words or solo words with their repeatedly counts. The tweet features will be a series of 1s and 0s. 1 represents that the Tweet contains N-gram, while 0 indicates its absence.

- *Unsupervised feature weighting methods*: Weighting techniques can be classified into two main classes: unsupervised and supervised techniques. The supervised technique utilizes previous data from the learning document to formulate a group of pre-produced classifies. This differ from the supervised feature weighting technique. Among the techniques utilized under unsupervised weighting are the TF-IDF (term frequency-inverse document frequency) and binary term frequency (TF).
- *Dimensionality reduction utilizing principal component analysis (PCA)*: It is considered a common method for extracting features. It has been applied in many fields and on wide and varied groups in various fields of biological and social sciences to the field of financing. As this approach is based on setting data points. To determine those points that are more important in that space, two criteria are applied: the proportion of variance and the Kaiser rule.
- *Word2vec model*: Word2Vec is used to create word embeddings. The models formed by using word2vec are little denotation two-layer neural networks. Once learned, they propagate semantic cases of words. The model takes a vast frame of text as a feed-in. It then constructs a vector scope that is usually of hundreds of dimensions. Each special word in the frame is allocated with symmetric vector in the scope. The words with common cases are placed in near closeness in vector scope. Word2vec uses one of the two constructions: continuous skip gram, which considered the current word is to forecast the neighboring window of case words. In this construction, the nearby case words are treat more constructions than words with outlying case. Or continuous bag of words (CBOW), the series of case words does not affect the prognosis as it is founded on bag of words sample.
- *Bag of words*: It is considered one of the simplest and most common methods of extracting features as this method is flexible. It is used to extract features from text data in various ways. Each word has a group of similar words; it is collected inside a bag called a word bag. WordPad is a display of textual data, which determines the frequency of words in a document. The word bag includes a dictionary of well-known words and the frequent presence of those words. The complexity of this model of feature extraction lies in the degree to which those words are present as well as how vocabulary is designed for those words. Despite the ease and flexibility of this model, word repetition is a problem that cannot be overlooked. The data with the highest frequency will be in control of the rest of the bag data. Higher frequency data may not be important, or model information may not be available. This problem is the main reason for ignoring related words.

- *TF-IDF*: In this way, we will be able to display unique words that carry the necessary information for a single document.
- *Opinion words and phrases:* They are phrases generally utilized to indicate opinions that are composed of bad or good, love or hate. This means some words indicate opinions devoid of utilizing opinion phrases.
- *Negations:* The existence of negative words might turn the opinion directing like not bad is equivalent to good.

3.4 Classification

After preprocessing and feature extraction level, we move to classification level. We pass the features into a classifier [10]. Many techniques were built for text classification. In this level, the general classes of techniques will be discussed, as well as their utilization specific to classification tasks. We understand that the discussed classes of techniques largely exist in other domains like categorical data or quantitative [24]. Sentiment classification strategies are classified into ML, lexicon-based approach, and hybrid technique as in [14]. In an ML approach, the popular ML algorithms are used in addition to the semantic indications of these algorithms, including naive Bayes classifier, SVM, decision tree, and others. We can classify machine learning into supervised learning and unsupervised methods. As for the lexicon-based approach, it is solely reliant on the dictionary of feelings, which is a collection of expressions of feelings previously collected. It is divided into two: the dictionary-based method and the corpus-based method. There is the hybrid method that combines the previous two approaches and through which we may get better results. Several optimization techniques can be used to optimize the classification process or feature selection processes [25, 26]. We will look at a number of widely used algorithms in Sect. 4 [14].

4 Sentiment Classification Techniques

In Fig. 2, we illustrate the classification technique. The general classes of techniques and their utilization for classification tasks will be further discussed.

4.1 Machine Learning Approach

An ML-based SA system was progressed in many earlier works to elicit public views in related topics. This system was capable of classifying tweets to various sentiment classes [27]. Text Classification Problem Definition: there is a list of training documents where each one is classified to a class. The classification model is suitable

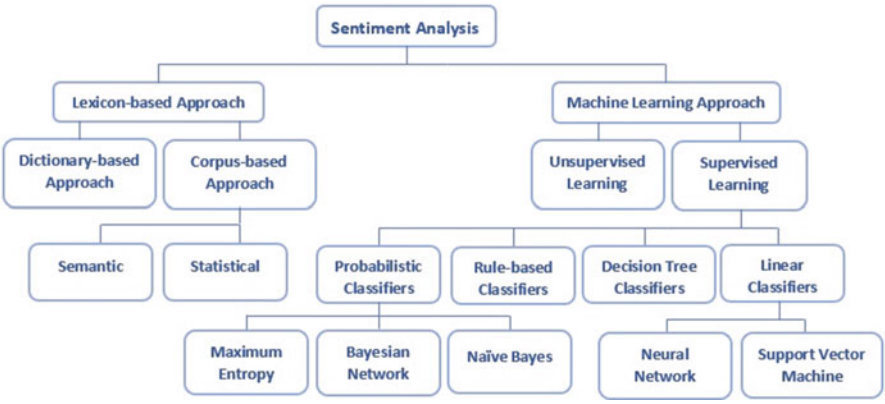

Fig. 2 Tree diagram of sentiment classification techniques

to the features in the inferior record to labeled according to the class label. Next for a given case of an undiscovered class, the model is utilized for the prediction a class label. The severe classification problem is in a situation when solely a single label was chosen to an occurrence. The machine learning technology uses many famous algorithms in machine learning and employs semantic functions [14].

4.1.1 Classification Based on Supervised Learning

Supervised learning is a type of ML [7, 28]. A supervised learning strategy relies on data called training data. Previously classified objects are entered into the device as training data. The device learns from that data. Then it will predict the unclassified data. There are many works under supervision; we will mention some of them [14, 29].

Naive Bayes Classifier

Naive Bayes is a simple, easy yet powerful rule suite [14]. Accordingly, it is used in both the training and classification stage [15]. It is a probabilistic classifier that uses Bayes' theorem to calculate the prospect of a tweet that belongs to a particular category like negative, positive, or neutral [20]. It can train the pattern of test on a group of documents that had been classified. The basic mechanism of the naive Bayes classifier is done by calculating the repetition of words concerning feelings in the message. Tweets are categorized and recorded according to the number of matches for emotional words, as the contents compare the word list to categorize documents into their correct category. The heaviness of the nodes is modified according to the significance of the tweets, and a more exact result can be generated for the rated feelings [2]. Preprocessed data is provided with the extracted feature as

an introduction to training the classifier by utilizing naive Bayes. As soon as the learning is done, in the classification process, it offers polarity of feelings. For instance, for review comment "I am Sad," it provides negative polarity as a result [15]. Naive Bayes classifiers are computational fast while making selections. Naive Bayes classifier is well used in real problems along with spam detection, SA, etc. It demands low processing memory and less execution time. The naive Bayes-type model estimates the backward potential classification based on the word distribution in the report [14].

Support Vector Machine (SVM)

Support vector machine (SVM) is a procedure for categorization of both nonlinear and linear data. SVM is generally used for text classification. If the data is linearly distinct, SVM studies the data, defines the selection limits, and utilizes the kernels for calculation that are executed in input space with the support vector machine.

Searches for the linear optimum split up hyperplane (the linear kernel) that is regarded as the vector, which splits up document vector in one class from the vectors in other class. Furthermore, all data regarded as a vector is categorized in a specific class. Hence, the work is to identify an edge between two categories that is distant from all documents [2, 15, 30]. It uses possibility space of linear functions in a loudly dimensional feature space, depending on kernel substitution. It will be learned with a training algorithm that executes a learning bias derived from statistical training theory. We can construct loudly nonlinear classification method by using SVMs [20]. This is done when the data is linearly close; the SVM utilizes nonlinear chart to turn data to a greater dimension. The problem is then fixed by locating a linear hyperplane [30]. SVM has the ability to replace the teaching styles dynamically each time there may be a new brand pattern throughout category [14].

Neural Network Classifiers

Neural networks (NN) are a specific set of algorithms, which have transformed machine learning [14]. Neural networks are stimulating with the aid of biotic neural networks [14]. The basic element in an NN is a unit or neuron. Hence, each unit takes a certain input i denoted by the vector X_i. Every neuron is also related with a group of weights A, which are utilized to calculate a function f of the inputs. A common function that is frequently utilized in the NN is the linear function stated as follows: $p_i = A * X_i$.

The question here is: how an NN may be utilized if all the categories cannot be carefully divided using linear separator? The categories might not be detached with the utilization of a single linear separator. The utilization of multilayers of neurons can be adopted to create a nonlinear classification limit. The purpose of such multilayers is to create multiple linear limitations that can be utilized to sacrificial bounded parts that belong to a specific class. In a network like this, the outputs of the

neurons in the previous layers are fed into the neurons in the next layers. The learning process of this kind of networks is rather sophisticated because the errors require to be back-propagated over varied layers. However, the general monitoring for text has been that linear categories mostly offer similar results to nonlinear data, and the refinements of nonlinear categorization methods are rather small [24]. Neural networks are criterion feature processes; that is why they can be used in almost any device training problem regarding studying a sophisticated mapping from the entry to the output area. Neural network-based planning has executed brilliant refinements in an expansion of herbal language processing involvement [14].

Random Forest (Decision Tree)

Decision forest provides a more accurate classification than a single decision tree [30] due to the fact that it consists of more than one decision tree [20]. The basis for decision tree is the hierarchical decomposition of data space, where the data is divided repeatedly until we find that each leaf node contains the minimum number of decisions [24]. Here, we will find that each tree will assign a specific category to each entry. The layer with the highest turn will be selected. Error rate depends on the strength of each tree separately from the forest, as well as the relationship between the trees in the forest itself. This means that reducing the error rate is contingent on the strength and independence of each tree in the forest [20]. To illustrate the work of the forest, we will take an illustrative example: X is the main forest, consisting of Y number of sub-trees. Each sub-tree is called X_i. The X_i is made up of branches with the same number of rows of the main X, which are the sample x with the substitution of X. By taking those samples with the substitution, this means that some of the traits in the original sample X may not be included in X_i, while it can be repeated in the others. Then the compiler builds a decision tree for each X_i. And at the end, we will have a forest with Y trees. To categorize an anonymous group, M, each tree brings back its own row prediction as a single vote. Therefore, the last decision of the M class is signed on which the most votes [30].

Rule-Based Classifier

Rules-based methods are based on entity recognition. In general, the nature of rule-based methods works as follows: At first, a group of rules is manually denoted or automatically trained [31]. The rules are adopted in the design of data space. The rule consists of two aspects. The left side called "pattern" represents the basic condition for the set of basic features. The pattern determines the regular expression based on features of tokens. The pattern matches a series of tokens, at which time the specified action is launched. The right side called "action" represents the category designation corresponding to the relevant feature. As for the action, it is possible to name a series of tokens as an entity, specify the beginning or ending designation for the entity, or specify a number of entities simultaneously [24, 31]. For instance, to point out any

series of symbols "Mr. Y" where Y is a capital letter as an individual entity, the subsequent rule can be specified: *(Token = "orthography" orthography type = FirstCap)→the person's name.*

The left side of the rule is a logical condition that can be expressed in DNF (disjunctive normal form). Nevertheless, in many circumstances, the condition on the left side is much plain and signifies a series of terms, which has to be existing in the document in order for the condition to be approved [24]. Generally utilized features to act tokens contain the token oneself, the part-of-speech (POS) identify the token, the orthography kind of the token, and if the token is inside some already defined gazetteer [31]. The lack of expression is hardly utilized, since such kind of rules is not probable but very factual for sparse text data, whereby many statements in the lexicon will normally not exist by default (sparseness property). The basic idea of making a method is to produce a group of rules, where all points are covered in an area of at least one base. An amount of criteria can be utilized to produce rules from learning data. The most common conditions used to create rules are trust and support [24].

Bayesian Classifiers

Bayesian networks (BNs), or belief networks (Bayes networks), are also called generative workbooks. It is one of the types of models with vector graphics. This means that the graphic links are represented by an arrow that indicates a certain direction. Attempting to construct a probability categorization depends on modeling keyword features in different categories. The idea of classifying texts is formed via the rear possibility of documents that belong to distinct categories, and this is based on the existence of a word in those documents. English language weights have an effect on the existence of words through many input groups. As the representations of the subject model are attractive, they get information that is absent in other methods. For instance, for documents that must be summarized, there will be an unambiguous representation as different documents to form that group. In usual ways, multiple document entries will be represented as a long text without differentiating the document limitations. Hence, for Bayesian classifiers, the rear possibilities are weighted through the cost of the category where the prediction is conducted [24, 32].

4.1.2 Classification Based on Unsupervised Learning

Unsupervised learning is a kind of machine learning. This type of machine learning relies heavily on speculative density in statistics. There is no goal in unsupervised learning, but it provides space for an evaluation of the model. The area can check a given model by relying on data such as the input values that are passed to the model [14, 29].

4.2 Lexicon-Based Approach

We mentioned in the previous sections the possibility of using the lexicon of feelings, which is one of the most important resources for most of the algorithms of sentiment analysis. Lexicon includes many words of opinion used in many classification tasks. Here, we will briefly mention some methods for creating lexicon of opinion words. Opinion words, polar words, or emotional words are utilized to express chosen situations if the opinion words are positive or unwanted if they are negative. Examples of positive opinion words are beautiful, good, etc. For negative opinion words, examples are bad, poor, etc. Words of opinion will not come at all times in the form of individual words; they can come in the form of opinion phrases or terms. For example, it costs a person an arm and a leg. Jointly, they are named the lexicon of opinion. Where it is important in the analysis of feelings. Words of opinion can be separated into two parts: the first is called the basic type, where all previous instances represent this type, and the second is called the comparative type. The comparative type is based on the principle of comparison and preference in opinion, for example, better, worse, and more. The preferential and comparative forms are used for the basic characteristics or conditions, for example, good and bad. Here, we will focus on the basic type. To compile a list of words of opinion, there are three main methods: the manual approach, the corpus-based approach, and the dictionary-based approach. The manual approach consumes a lot of time, so it is usually not used; in some cases, it is combined with automated methods. Here, we will address the two automated approaches [14, 33, 34]:

- Dictionary-Based Approach

  To obtain polarity of sentiments, a lexicon of opinion is used. The polarity number of negative and positive words shown for each tweet is calculated. This method determines the highest number of polarities. If polarity is equal to both positive and negative, then polarity is considered neutral to that tweet. The purpose of using this method is to facilitate the access to sentiment words with their directions. However, it failed to define the directions that adopt the context formula for sentiment words [6]. Words of opinion are gained manually to create a group of opinion words. This group can be expanded with the help of searching within WordNet to add new opinion words or a list of their synonyms or oppositions. Then the examination is done manually to remove and correct errors. The downside of this type is its weakness in identifying words of opinion in context-specific directives [14, 34].
- Corpus-Based Approach

  In this model, after extracting opinion words from tweets, their direction is determined. Words of opinion will be a mixture of verbs and adjectives in addition to circumstances. Conditions and verbs are not adopted, and to calculate their direction, the dictionary-based method is used. As for adjectives, an adjective is a word used to describe and qualify an object. It is field dependent, where the corpus-based method will be utilized to obtain the semantic direction of the adjectives [30, 34].

5 Challenges Involved in Sentiment Analysis

There are some challenges that should be faced in sentiment analysis. Some of them are listed as follows [14]:

- *Language problem*: The English language is used to analyze sentiment well due to the availability of resources in English that facilitate the analysis process. Resources are lexicons and dictionaries. There are many researchers interested in analyzing sentiment in other languages, including Chinese, Arabic, German, and others. This makes it challenging for researchers to create resources in other languages, i.e., glossary and dictionaries.
- *Natural language processing (NLP)*: The utilization of NLP requires further improvements in the sentiment analysis process, as it has become a magnet for researchers. Natural language processing offers better mining results and also provides a good awareness of the language. Mining for opinion is context based or field based. Opinion mining needs to pay a lot of attention to it because field-based mining provides a good result than context-based mining. Domain-based mining is complicated or more difficult to develop.
- *Fake opinion*: Fake opinion, or fake review, refers to false reviews that mislead consumers by presenting opinions that are not real, negative or positive, in order to reduce the condition of the object. Such SPAM makes sentimentality ineffective in many applications.

6 Conclusion and Future Work

The data that is used to examine sentiment is social networks. Twitter is the most important of them. It is analyzed in different perspectives to express sentiment and opinions. We explain the concept of sentiment and the structure of data processing to extract opinion at various levels of sentiment analysis. The polarity of sentiment was also categorized as negative, positive, and neutral. In addition to subcategories, which are very negative and very positive. Many approaches have been used in the field of SA, including ML and lexicon. The dictionary of abbreviations and emoticons is also used in sentiment analysis. Here, we presented an overview of the most important algorithms that had good results. Naive Bayes and SVM algorithms are commonly utilized to identify the problem of classification of sentiment. SA is a broad field, opens wide means for research fields and various issues. So the field of investment has become large in the SA. Where it is adopted at the political and economic level and many areas. However, there are some challenges facing the SA, one of the most vital of which is limited data resources for non-English languages, as explained in Sect. 5.

Sentiment analysis is a broad and important area. Many institutions in various fields are adopting sentiment analysis in their work. Millions of tweets every day bring up various topics. This diversity in the content of the tweets needs to be

analyzed according to the direction of the field or topic that was raised. There are many methods and algorithms used in the analysis and approved to reach an opinion that is closer to the truth or accuracy in the results. All studies presented dealt with textual sentiment or textual data. The tweets can contain opinions that are presented in the form of pictures or videos or can be in the form of a link to communicate the sentiment or opinions more accurately and clearly. So we recommend that there be studies to analyze the sentiment of images or videos, or possible links, which are an important element in communicating the sentiment directly because of the important feelings that it holds for the opinion.

References

1. R. Wagh, P. Punde, Survey on Sentiment Analysis Using Twitter Dataset, in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, (IEEE, 2018), pp. 208–211
2. B.S. Dattu, D.V. Gore, A survey on sentiment analysis on twitter data using different techniques. Int. J. Comp. Sci. Inform. Technol. **6**(6), 5358–5362 (2015)
3. H. Hajipoor et al., A survey on twitter sentiment analysis. Proceedings of the First International Conference on Web Research (ICWR), Tehran, Iran, 15–16 (2015)
4. G. Beigi, X. Hu, R. Maciejewski, H. Liu, An Overview of Sentiment Analysis in Social Media and Its Applications in Disaster Relief, in *Sentiment Analysis and Ontology Engineering*, (Springer, Boston, MA, 2016), pp. 313–340
5. R. Varghese, M. Jayasree, A survey on sentiment analysis and opinion mining. Int. J. Res. Eng. Technol. **2**(11), 312–317 (2013)
6. A.P. Jain, V.D. Katkar, Sentiments Analysis of Twitter Data Using Data Mining, in *2015 International Conference on Information Processing (ICIP)*, (IEEE, 2015), pp. 807–810
7. L. Abualigah, A.H. Gandomi, M.A. Elaziz, H.A. Hamad, M. Omari, M. Alshinwan, A.M. Khasawneh, Advances in meta-heuristic optimization algorithms in big data text clustering. Electronics **10**(2), 101 (2021)
8. A. Kumar, T.M. Sebastian, Sentiment analysis on twitter. Int. J. Comp. Sci. Issues (IJCSI) **9**(4), 372 (2012)
9. N. Zainuddin, A. Selamat, R. Ibrahim, Hybrid sentiment classification on twitter aspect-based sentiment analysis. Appl. Intell. **48**(5), 1218–1232 (2018)
10. A. Dalmia, M. Gupta, V. Varma, IIIT-H at SemEval 2015: Twitter sentiment analysis–the good, the bad and the neutral! Proc. 9th Int. Worksh. Seman. Eval. (SemEval) **2015**, 520–526 (2015)
11. R. Pandarachalil, S. Sendhilkumar, G.S. Mahalakshmi, Twitter sentiment analysis for large-scale data: an unsupervised approach. Cogn. Comput. **7**(2), 254–262 (2015)
12. V. Carchiolo, A. Longheu, M. Malgeri, Using Twitter Data and Sentiment Analysis to Study Diseases Dynamics, in *International Conference on Information Technology in Bio-and Medical Informatics*, (Springer, Cham, 2015)
13. A. Giachanou, F. Crestani, Like it or not: a survey of twitter sentiment analysis methods. ACM Comput. Sur. (CSUR) **49**(2), 1–41 (2016)
14. C. Bhagat, D. Mane, Survey on text categorization using sentiment analysis. Int. J. Sci. Technol. Res. **8**(8), 1189–1195 (2019)
15. G. Gautam, D. Yadav, Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis, in *2014 Seventh International Conference on Contemporary Computing (IC3)*, (IEEE, 2014)
16. L.M.Q. Abualigah, E.S. Hanandeh, Applying genetic algorithms to information retrieval using vector space model. Int. J. Comp. Sci. Eng. Appl. **5**(1), 19 (2015)

17. L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J. Supercomput. **73**(11), 4773–4795 (2017)
18. L. Abualigah, M. Qasim, *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering* (Springer, Berlin, 2019)
19. V.S. Pagolu et al., Sentiment Analysis of Twitter Data for Predicting Stock Market Movements, in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, (IEEE, 2016)
20. B. Gokulakrishnan, P. Priyanthan, T. Ragavan, N. Prasath, A.S. Perera, Opinion Mining and Sentiment Analysis on a Twitter Data Stream, in *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*, (IEEE, 2012), pp. 182–188
21. A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R.J. Passonneau, Sentiment analysis of twitter data. Proc. Worksh. Lang. Soc. Media (LSM) **2011**, 30–38 (2011)
22. A.G. Shirbhate, S.N. Deshmukh, Feature extraction for sentiment classification on twitter data. Int. J. Sci. Res. (IJSR) **5**(2), 2183–2189 (2016)
23. R.N. Waykole, A. Thakare, A review of feature extraction methods for text classification. IJAERD **5**(04), 351–354 (2018)
24. C. C. Aggarwal, C. X. Zhai (eds.), *Mining Text Data* (Springer Science & Business Media, New York, 2012)
25. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm. Comput. Methods Appl. Mech. Eng. **376**, 113609 (2021)
26. L. Abualigah, A. Diabat, Advances in Sine Cosine Algorithm: A comprehensive survey. Artif. Intell. Rev. **54**, 1–42
27. J. Du et al., Leveraging machine learning-based approaches to assess human papillomavirus vaccination sentiment trends with twitter data. BMC Med. Inform. Decis. Mak. **17**(2), 69 (2017)
28. L. Abualigah, A.H. Gandomi, M.A. Elaziz, A.G. Hussien, A.M. Khasawneh, M. Alshinwan, E.H. Houssein, Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis. Algorithms **13**(12), 345 (2020)
29. T. Upadhya, S. Raj, S. Pathak, Machine learning techniques for code optimization. Mach. Learn. **6**(07) (2019)
30. X. Fang, J. Zhan, Sentiment analysis using product review data. J. Big Data **2**(1), 1–14 (2015)
31. J. Jiang, Information Extraction from Text, in *Mining Text Data*, (Springer, Boston, MA, 2012), pp. 11–41
32. A. Nenkova, K. McKeown, A Survey of Text Summarization Techniques, in *Mining Text Data*, (Springer, Boston, MA, 2012), pp. 43–76
33. R. Feldman, Techniques and applications for sentiment analysis. Commun. ACM **56**(4), 82–89 (2013)
34. B. Liu, Sentiment analysis and subjectivity. Handb. Nat. Lang. Process. **2**(2010), 627–666 (2010)

Improving Automated Arabic Essay Questions Grading Based on Microsoft Word Dictionary

Muath M. Hailat, Mohammed A. Otair, Laith Abualigah, Essam H. Houssein, and Canan Batur Şahin

1 Introduction

With computer technology development in the educational field, an increasing number of students and a growing educational community, traditional and electronic, giving rise to automatic essay scoring systems by comparing and evaluating the student answer with the model answer, have been noticed. It is easy to implement automatic grading system for multiple choice questions, true and false questions, and fill-in-the-blank questions because the answers are specified compared with the answers of essay questions [1].

Essay questions are more challenging in comparison to other types of questions. This is because they need more time for grading, but they are preferable as cheating will not be easy anymore [2]. The researchers started in automatic assessment since the 1960s of the last century where the first model developed is PEG (Project Essay Grade) in 1960s, IEA (Intelligent Essay Assessor) was developed in 1997, E-Rater (Electronic Essay Rater) was developed in 1998, and IntelMetric system was

M. M. Hailat · M. A. Otair
Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

L. Abualigah (✉)
Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia

E. H. Houssein
Department of Computers and Information, Minia University, Minia, Egypt

C. B. Şahin
Department of Engineering and Natural Sciences, Malatya Turgut Ozal University, Malatya, Turkey

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_2

developed in 1998 [3]. It is found that British teachers spend 30% of their time in correcting answers and that estimate is worth 3 billion pounds yearly. So automatic grading solves manual grading process problems like high cost and time-consuming task, with benefits for both the teacher and the student, where the teacher saves time and answers are more accurate. On the other hand, students can know their grade directly [4].

Automated grading system is widely used by schools, universities, companies, or any entities that use online exams such as TOEFL [5]. Also, it is described as an efficient and effective manner for grading essay questions in the form of numeric or letter grade [6]. Automatic Arabic grading technique contains two stages for essay grading. Firstly, prepare useful text for processing and analyzing after putting it in an automatic grading system. Preprocessing text includes normalization text by removing diacritics and special characters, along with replacing the shape of some letters: /Alef/ (أ) will be /Ah/ (ا), /tah/ (ة) will be /ha'/(ه), and /hamzah/ (ئ) will be /ya'/ (ي).

Then, use stop word removal by identifying prepositions and pronouns. In addition, identify some words that can be considered irrelevant to semantic issues [7, 8]. After that, use stemmer to retrieve the root of words, which makes searching more efficient. The next step will be using Microsoft Word dictionary for word synonyms, which is useful when different words are used in the student answer, which have the same meaning with teacher answers.

Secondly, use naive Bayes classifier to classify the answers of the student. Next, use inner product similarity in order to measure relevancy between student answers and teacher answers to retrieve the score. Due to large data and the rapid development of Internet content, text classification was developed to effectively manage and utilize large amount of data [9].

In classification, there are many types of classifiers used for automatic essay grading system like KNN, SVM (support vector machine), and NB (naive Bayes classifier). Naive Bayes will be used in this chapter, and it is known as a simple classifier since it is very easy to implement and very fast [10]. Similarity measures are very important for creating intelligent systems that exhibit behavior similar to humans. Also, text similarity gets approximate of human similarity related to language [11].

There are many types of text similarities. One of these types is inner product, which is occasionally called scalar product or dot product [12]; in this chapter, it is utilized for the purpose of retrieving the student answer that is more relevant to the teacher answer. Many techniques are used in Arabic essay grading system such as system using text similarity algorithms [2] and system using statistical besides computational linguistic technique [5]. In this chapter, naive Bayes classifier and inner product similarity are applied to classify and measure similarity between teacher answers and student answers in order to find the final score. In the future, optimization method can be used [13].

2 Literature Review

This section presents the perspective of literature review and related works; it presents an overview about Arabic essay grading techniques, and the main topics of Arabic essay questions grading techniques include text similarity, text classification, and MS dictionary.

2.1 Text Classifications

Text classification is considered as a very important field that is used to classify documents depending on predefined groups. Most text classification techniques are used to classify English documents. Unfortunately, few classification techniques were applied to the Arabic documents [14]. Text classification indicates to building automatically system to classify multi-documents to its right categories. This system is known as knowledge engineer technique. For example, when there are many documents, we need to classify them to their correct class as "sport," "politics," or "art." Then, the classification model or system will be able to classify each document to its correct class; if one document belongs to more than one class, this is called multilabel document. If the document belongs to only one class, it is called single-label document [15].

2.2 Text Classification Algorithms

Text classification, known as topic spotting, is defined as the process for deciding whether the set of data belongs to a predefined class or not [16–19]. This task belongs to IR (information retrieval) and ML (machine learning) fields, which has the attention of many developers and researchers. According to the large number of automatic classifications is required, which can classify text automatically to predefined classes depending on their content, and automatic classifications have many advantages such as being very fast and accurate compared with manual classifications. Most researchers work on English language text, but a limited number of researchers work on Arabic language text [20]. Text classifications involve many applications like automated essay grading, filing patent into patent records, and spam filtering [21]. The key purpose of text classification algorithms is to reduce information loss while maximizing reduction dimensionally [22]. Text classifications techniques include the following classifiers: NB, SVM, KNN, Rocchio classifier, linear classifier, and decision tree (DT) classifier.

2.2.1 Naive Bayes (NB)

It is a classification technique that is based on Bayesian theorem which belongs to a probability classifier and is considered simple as a probabilistic classifier as well as a powerful, simple theorem [14] and is useful for large data. Naive Bayes classifier is widely used for automated essay grading with multi-language. This chapter uses NB classifier with Arabic essay question grading.

The main formula of naive Bayes is shown in Eq. (1):

$$P(C/D) = \frac{P(C).P(D/C)}{P(D)} \tag{1}$$

where $P(C/D)$: probabilities that document D belongs to predict category C.
$P(D)$: document probability
$P(C)$: category probability
(D/C): probabilities of document D given category C

Figure 1 presents the training steps to build naive Bayes classifier and classification process [23].

Some learning algorithms are applied to train a classifier based on the input data. With the results of trained classifier applied to any new data, labelling process can be done. In this chapter, 80 answers are used as training data set where in these steps two features are more relevant and descriptive to the classification task extracted, fail or good.

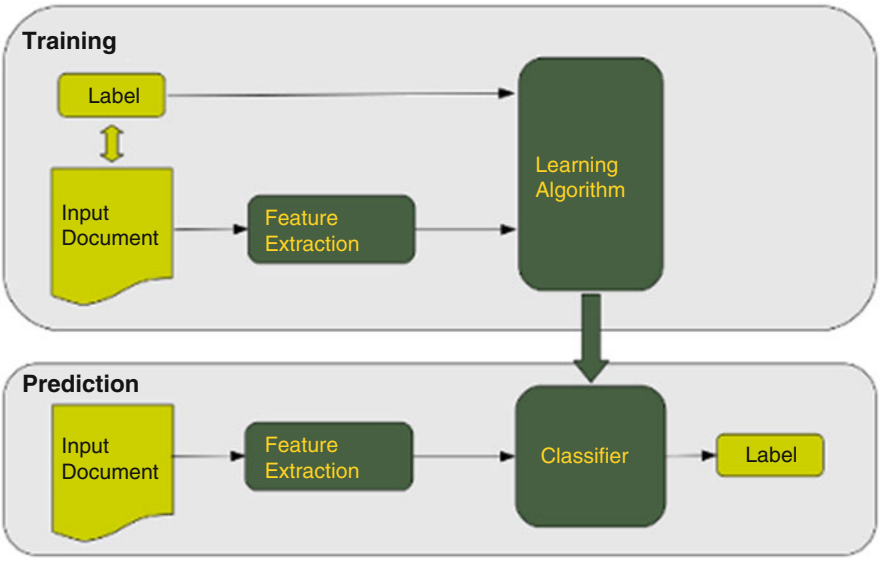

Fig. 1 Built and process naive Bayes classifier

NB algorithm is widely utilized in text classification, which has shown good results and high performance. NB is known as a popular machine-learning algorithm because of its efficiency and optimal time complexity. Naive Bayes presents two popular models: multivariate Bernoulli naive Bayes classification and multinomial naive Bayes classification. Naive Bayes multinomial model is preferred to be used when database is large, and it is more suitable for text files. On the other hand [15], multinomial focuses on the number of times the words frequent in the document, which tells about the words occurring in the document and term frequency (TF). Bernoulli model focuses on binary concept that indicates the amount of times a word frequents in the document. However, it is not as multinomial Naïve Bayes; it does not tell about TF [24].

2.2.2 Support Vector Machine (SVM)

SVM is a supervised machine learning technique that can be utilized for text classifications, along with automated essay grading. It needs training set positive and negative that has no need for other classification method used in dimensional space for the purpose of separating between negative and positive data seeking decision surface, called hyperplane. The closest document represents a decision surface called support vector [15]. Figure 2 presents SVM using hyperplane [25].

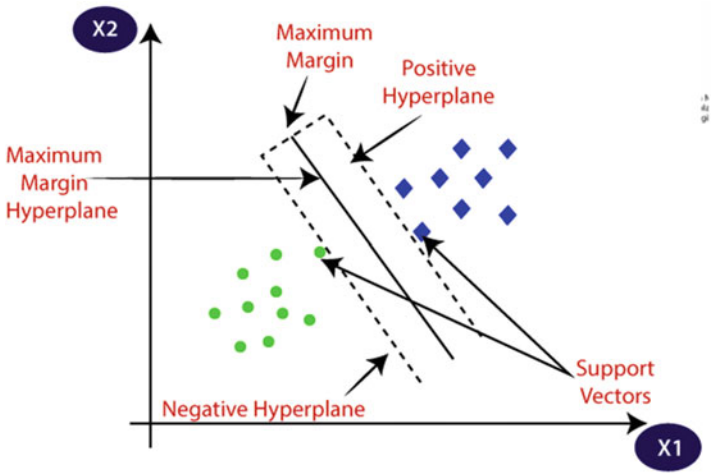

Fig. 2 SVM using hyperplane

2.3 Text Similarity (TS)

Text similarity plays a major role in text applications such as information retrieval, text classifications, short answer questions, automated essay questions, and answer scoring [26, 27]. Finding similarities between words, paragraphs, and documents is important in text similarities [28].There are three approaches of text similarities:

1. String-based similarity: this similarity helps in measuring the structure of characters by two ways, term-based similarity and character-based similarity.
2. Corpus-based similarity: this similarity helps in measuring the similarity among words depending on information collected from huge corpora.
3. Knowledge-based similarity: this similarity helps in measuring the similarity among words utilizing information derived from a network such as WordNet that considers huge database for English verb, noun, adverb, and adjective [29]. In this chapter, inner product similarity will be used to score and grade students' answers.

2.3.1 Inner Product Similarity (Dot Product)

Similarity measure computes the level of similarity among a duo of vectors, since documents and queries are both vectors. A similarity measure denotes the similarity among two queries, two documents, or even one document and one query. Inner product similarity measures similarity between documents and query as formula product:

$$\sum_{k=1}^{t} (dik \bullet qk)$$

where d_{ik} is the weight of term i in document k and q_k is the weight of term i in the query. For weighted vector term, inner product similarity is Sumption of the products with weights of the matched terms.
For binary vectors, inner product similarity measure is the number of matched queries in the document. Below is an example for binary vectors:

$$\text{Document} = 1,1,1,0,1,1,0 \text{Query} = 1,0,1,0,0,1,1 \text{sim}\,(\text{D},\text{Q}) = 3$$

where D is the document and Q is the query.

2.4 Evaluation Metrics

Evaluation results of the proposed technique in this paper by comparing results between human score and automated score measures with the use of MAE (mean absolute error) and PCR (Pearson correlation result).

2.4.1 Mean Absolute Error (MAE)

MAE is used for evaluating scores between human scores and automated scores for students' answers. MAE result is calculated according to Eq. (2):

$$\mathrm{MAE} = \frac{1}{N}\sum_{I=1}^{N} Xi - Yi \tag{2}$$

where $\boldsymbol{x}$ is the human score, $\boldsymbol{y}$ is the automated score, and $\boldsymbol{N}$ is the number of data test.

2.4.2 Pearson Correlation Result (PCR)

Pearson correlation measures the strength correlation between human scores and automated scores for students' answers. PCR result is calculated according to Eq. (3).

$$\mathrm{PCR} = \frac{\sum \boldsymbol{xy}.\frac{\sum \boldsymbol{x}.\sum \boldsymbol{y}}{N}}{\sqrt{\sum \boldsymbol{x^2} - \frac{\sum \boldsymbol{x^2}}{N}.\sum \boldsymbol{Y^2} - \frac{\sum \boldsymbol{Y^2}}{N}}} \tag{3}$$

where $\boldsymbol{x}$ is the human score, $\boldsymbol{y}$ is the automated score, and $\boldsymbol{N}$ is the number of question test.

2.5 Previous Related Studies

In [30], an automated Arabic essay grading system based on SVM and text similarity algorithm is proposed; this thesis presented two main processes: firstly, using WordNet to find all possible meanings of the given words and, secondly, using feature extraction. In addition, using cosine similarity to find the similarity level among student answer and teacher answer. The dataset collected from computer, social and science books that contain 40 questions with three class of answers as

120 questions. This thesis shows that automatic Arabic essay grading using WordNet gives a better result without the use of WordNet.

In [2], an automatic Arabic essay grading system based on text similarity was presented. The study used 210 short answers as dataset to apply string-based similarity (Damerau-Levenshtein algorithm, N-gram algorithm) and corpus-based algorithms (LSA, DISCO) separately besides comparing the result and using preprocessing data steps: *firstly*, using raw then tokenization and, *secondly*, stop word removal. After that, stemming and stop stem. The study proves that using similarity algorithm gives effective solution that helps the teacher to use automated grading system with high precision.

In [31], they presented an automated assessment of school children essay in Arabic. The study used data set over three hundred students in Riyadh, "the capital of Saudi Arabia," applying preprocessing technique and using hybrid technique that combines rhetorical structure theory (RST) with latent semantic analysis (LSA); the experiment of their system showed an accuracy of 78.33% and a correlation of 0.79 with teacher evaluation.

In [32], an approach for automated scoring for Arabic short answer essay questions is suggested. This study utilized cosine similarity to measure similarity between model answer and students' answer. The data set of this study examines one exam with 11 questions and one model answer for individual question. Furthermore, the phase of proposed approach as: firstly, extracted keywords for student answer and model answer and then used synonymous for extracted words in student and model answer to have accurate result. After that, use cosine similarity to measure the accuracy of model answer and student answer. The result of using this approach given correlation equals to 95.4% and 84.5 as correct ratio.

In [33], the researcher introduced an automated system for essay scoring of online exams in Arabic. This study presented an automated grading system using heavy stemming approach and light stemming approach. Firstly, use heavy stemming approach for both student answer and model answer, and then start using heavy stemming by removing numbers, diacritics, and any letter from other language. Next, applying processing word by remove stop words, remove "AL." Thenceforth, normalize the word and remove suffix or prefix if the word has more than three letters. Lastly, apply similarity to each word in student answer and model answer. Secondly, use light stemming by removing prefix and suffix and then applying similarity between student answer and model answer. The proposed technique has an effective result for Arabic essay questions.

In [34], the authors proposed a hybrid method for automatic Arabic essay scoring. In this study, five phases were adopted: (1) preparing data set, which consists of 610 answers; (2) preprocessing data using normalizing, tokenization, and stemming; (3) finding the synonym of words using Arabic WordNet; (4) applying latent semantic analysis with TF-IDF and cosine similarity; and (5) applying modified LSA using part of speech (POS), which assigns words to fixed part of speech (noun, verb, adjective, and adverb). Using modified LSA improves the accuracy of Arabic essay scoring.

In [1], the authors proposed an Arabic short answer scoring method with effective feedback for students. This study used 14 string-based and two corpus-based similarities as well as a comparison between them. After that, evaluating and using combination of these similarity measures. In this research, the Arabic dataset was collected containing 50 questions together with 12 answers ends with a total of 600 answers. Applying four methods raw, stop, stem, and stop stem to deal with string-based similarity. In addition to two model holistic, partitioning to deal with corpus-based similarity, this study presented using this system to implement in real scoring environment.

In [35], short answer grading utilizing string similarity and corpus-based similarity was introduced. The system works at three phases: firstly, measuring the similarity between model answers and student answers using string-based algorithms using four models: stop, raw, stop stem, and stem; secondly, using corpus-based similarity DISCO1 and DISCO2 algorithms using stop word removal, getting distinct along with building similarity matrix; and, thirdly, merging the similarity values of string-based algorithm with values of corpus-based algorithms. The dataset was collected from the University of North Texas containing 80 questions and 2273 student answers. The system gets result 0.504 from combined N-gram with DISCO1.

In [4], they proposed an automated assessment of student's Arabic free-text answers. In this study, latent semantic analysis (LSA) technique was used. A set of 29 answer papers as dataset was collected from "System Designing" course in March 2011, which was written in Arabic language. The answer will be added to the system, and several tasks are performed such as cleaning the text, normalizing letter, spell-checking words, stemming, processing synonyms, and removing stop words. Next, word-by-context matrix (WCM) is created. After that, it calculates the weight. Then, it calculates a cosine similarity to compare between student answer and referential answer. At last, calculating the correlation between human grading score and system score that resulted 0.91. However, the correlation between human score and LSA algorithm was 0.88.

3 Proposed Model

This chapter proposes a technique that aims to improve an automatic Arabic essay grading technique. As presented in Fig. 3, the process starts by preprocessing phase, where tokenization step divides answers for small pieces of tokens. For normalization step, it is used to replace special letter shapes and remove diacritics. Then, stop word removal step removes meaningless and useless words. Lastly, stemming process is used to get the stem and root of the words. All the preprocessing phase is meant to be implemented for both student answer and dataset (corpus). Then, classifying by naive Bayes classifier to get accurate result also for both students answers among with dataset (corpus). After that, Microsoft Word dictionary is used to compare and get enough synonyms for both students' answers and model answers in order to get better results. Finally, showing results with the use of inner product

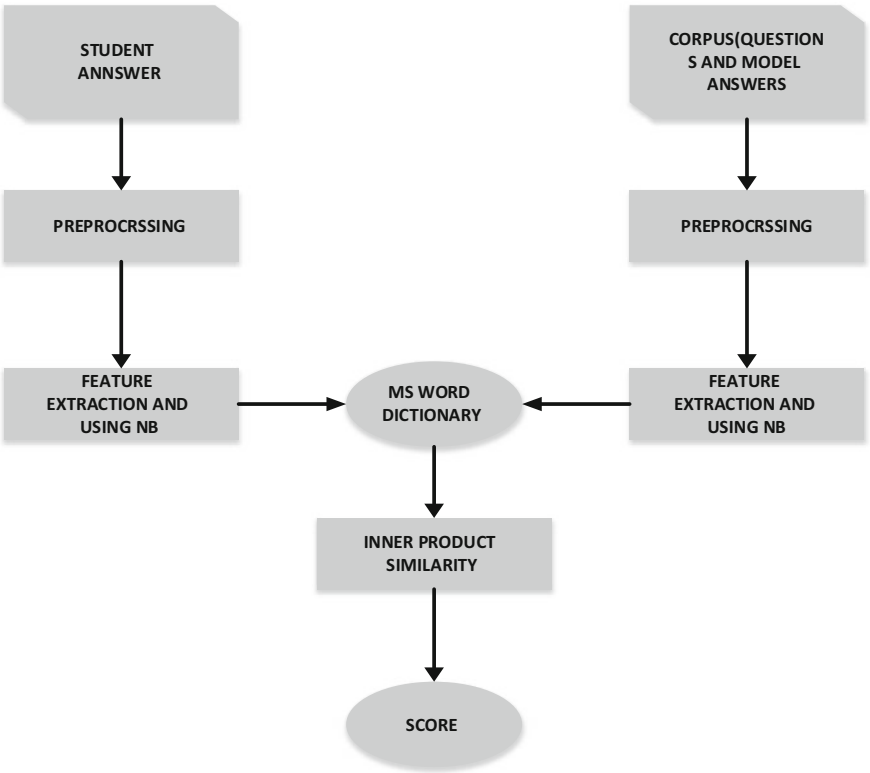

Fig. 3 Diagram of the proposed technique

similarity then compare the results showed by inner product similarity with human score results so the evaluation among with the efficiency of the proposed technique can be measured.

3.1 Preprocessing

The phase is used for converting text from difficult one to much easier one to handle. Also, transformed text is meant to be in vector format [36]. Furthermore, implementing word extraction to its clear format, applying tokenization, normalization, stop word removal and stemming that reduce ambiguity of words to increase efficiency. This step is considered very important in classification process especially when using highly inflected language such as Arabic language to support and increase result efficiency [37]. Figure 4 shows the preprocessing steps.

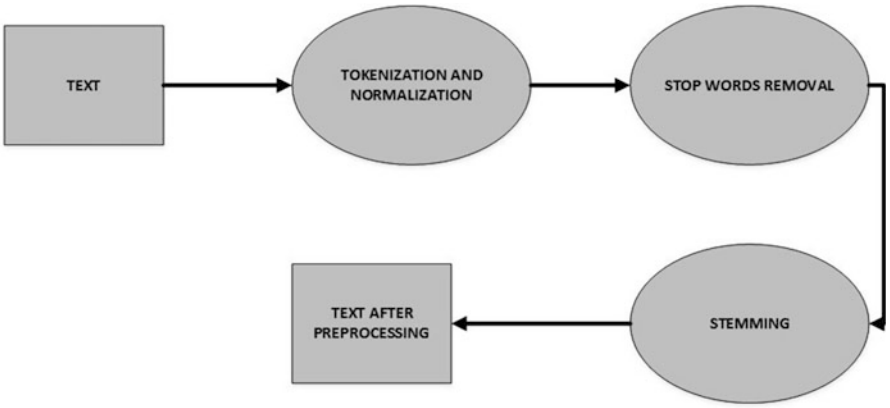

Fig. 4 Preprocessing steps

3.1.1 Tokenization and Normalization Processes

Tokenization is the first step in preprocessing phase, which divides students' answers in addition to model answers into small pieces and number of strings. Some examples of tokenization remove stop marks are (",", "."، "?") from students answers and model answers. Then, divide sentences into tokens:

Text: "هي عملية ابقاء المعلومات تحت سيطرتك الكاملة"

but when converting this text into tokens, it will be

"'هي', 'عملية', 'ابقاء', 'المعلومات', 'تحت', 'سيطرتك', 'الكاملة'".

Normalization is a remarkable step in preprocessing phase for the purpose of achieving best results. The step of Normalization's intent is replacing the shape of characters besides transforming into single form and also removing diacritics and special characters.

3.1.2 Stop Word Removal

Stop word removal is defined as removing unnecessary and meaningless words that are useless and give unimportant meaning in text such as pronouns, conjunctions, and prepositions [38, 39]. The process applies both students' answers and model answers after tokenization step. Stop words listed in table to get removed after tokenization step. For example, 'أي', 'أين', 'أيها', 'إذ', 'إذا', 'إذما', 'إذن', 'أما', 'إما', 'إن', 'إنا', 'إنما', 'إنه', 'إي', 'إيه', 'إلا', 'إلى', 'إليك', 'إليكم', 'إليكما', 'إليكن', 'إم', 'إن', 'أنا', 'أنت', 'أنتما', 'أنتن', 'أنتم', 'أنتما', 'أنتن', 'آه', 'آها', 'آهاها', 'أو', 'أولاء', 'أولالك', 'أوه', 'آي', 'أيا', 'أيها', 'أيهم', 'أيهما', 'أيهن', 'بخ', 'بس', 'بعد', 'بعض', 'بك', 'بكم', 'بكما', 'بكن', 'بل', 'بلى', 'بما', 'بماذا', 'بمن', 'بنا', 'به', 'بها', 'بهم', 'بهما', 'بهن', 'بي', 'بين', 'بيد', 'تلك', 'تلكم', 'تلكما', 'ته', 'تي', 'تين', 'تينك', 'ثم', 'ثمة', 'حاشا', 'حبذا', 'حتى', 'حيث', 'حيثما', 'حين', 'خلا', 'دون', 'ذا', 'ذات', 'ذاك', 'ذان', 'ذانك', 'ذلك', 'ذلكم', 'ذلكما', 'ذلكن', 'ذه', 'ذو', 'ذوا', 'ذواتا', 'ذواتي', 'ذي', 'ذين', 'ذينك', 'ريث', 'سبحان', 'سرعان', 'سقى', 'سوف', 'سوى', 'سواء', 'شبه', 'صه', 'صهٍ', 'طاق', 'طالما', 'طفق', 'طق', 'ظل', 'عا', 'عدا', 'عسى', 'عل', 'على', 'عليك', 'عليه', 'عما', 'عن', 'عند', 'غير', 'فإذا', 'فإن', 'فلا', 'فمن', 'في', 'فيم', 'فيما', 'فيه', 'فيها', 'قد', 'قط', 'قلما', 'قليلا', 'كأن', 'كأنما', 'كأي', 'كأين', 'كذا', 'كذلك', 'كل', 'كلا', 'كلاهما', 'كلتا', 'كلما', 'كليكما', 'كليهما', 'كم', 'كما', 'كي', 'كيت', 'كيف', 'كيفما', 'لا', 'لاسيما', 'لبيك', 'لدن', 'لست', 'لستم', 'لستما', 'لستن', 'لسن', 'لسنا', 'لعل', 'لك', 'لكم', 'لكما', 'لكن', 'لكنما', 'لكي', 'لكيلا', 'لم', 'لما', 'لن', 'لنا', 'له', 'لها', 'لهم', 'لهما', 'لهن', 'لو', 'لولا', 'لوما', 'لي', 'لئن', 'ليت', 'ليس', 'ليسا', 'ليست', 'ليستا', 'ليسوا', 'ما', 'ماذا', 'متى', 'مذ', 'مع', 'معاذ', 'مكانكم', 'مكانكما', 'مكانكن', 'مه', 'مهما', 'من', 'منذ', 'منه', 'مهما', 'نحن', 'نحو', 'نعم', 'ها', 'هاتا', 'هاتان', 'هاته', 'هاتي', 'هاتين', 'هاك', 'هاكم', 'هاكما', 'هاكن', 'هبك', 'هبكم', 'هبكما', 'هبكن', 'هذا', 'هذان', 'هذه', 'هذي', 'هذين', 'هكذا', 'هل', 'هلا', 'هم', 'هما', 'هن', 'هنا', 'هناك', 'هنالك', 'هو', 'هؤلاء', 'هي', 'هيا', 'هيت', 'هيا'; all of these words are going to be deleted after applying stop word removal step.

As an example, in this paper:

Text "هي عملية ابقاء المعلومات تحت سيطرتك الكاملة" as token "'هي', 'عملية', 'ابقاء', 'المعلومات', 'تحت', 'سيطرتك', 'الكاملة'"

after applying process stop word removal, the text becomes

"'عملية', 'ابقاء', 'المعلومات', 'تحت', 'سيطرتك', 'الكاملة'"

3.1.3 Stemming

Stemming is a method to minimize a word to get its root or stem, which is mostly used in information retrieval to increase recall rate [40, 41]. The main process of stemming is to find the word root or stem, which means removing all the affixes of a given word. Arabic stemmer is grouped into two main types. First, light stemmer gets the root by removing suffixes, prefixes, and infixes to find the root such as snowball stemmer. Second, heavy root stemmer is used to find the root by reducing affixes and transforming some word letter in order to find the stem or root such as Khoja stemmer and ISRI stemmer [42]. Arabic language is deliberated as one of the most challenging to retrieve word's root for the reason that it has many morphological variations [43]. After the normalization process of the words is done, this leads to the last process which is stemming process. Arabic text classification uses multi-types of stemmers. The ISRI (Information Science Research Institute), which is called Arabic stemmer, can be benefited from it to get the stem and root of the words [26]. ISRI stemmer uses root dictionary for the purpose of getting the stem and root for retrieval task information, removing diacritics, and normalizing some letter shapes. Also, removing length two or three prefixes, removing connectors. In addition to normalizing Alif (change أ to ا), if the word equals three letters, it returns as stem, with a consideration of four cases while depending on the length of word (length = 4, length = 5, length = 6, length = 7) [44]. For example: Text "هي عملية ابقاء المعلومات تحت سيطرتك الكاملة" as token "'هي', 'عملية', 'ابقاء', 'المعلومات', 'تحت', 'سيطرتك', 'الكاملة'" as stop word removal "'عملية', 'ابقاء', 'المعلومات', 'تحت', 'سيطرتك', 'الكاملة'" as stemming "'عمل', 'بقاء', 'علم', 'تحت', 'المعلومات', 'تحت', 'سيطرتك', 'الكاملة'"

3.2 *Naive Bayes Classifier*

NB classifier is a simple probabilistic classifier where each feature is treated independently. NB has many different types such as multinomial, Bernoulli, and Gaussian type. NB works well in many fields such as text classification. NB is considered fast and easy as needed for implementation, so that it can be used as baseline in TC [24]. In [45], a comparison among many types of text classification is presented. Correspondingly, it is an indication for Naïve Bayes classifier to be the best for document classification by using preprocessing and feature extraction methods that get better mining quality performance. In addition, quality of data that affected

performance of mining results. In this chapter, naive Bayes phase comes after preprocessing phase, which is compared with the most advanced text classification method called support vector machine (SVM). Later, compare the three types of naive Bayes with one other. Then, select the best among the three types and use Gaussian naive Bayes classifier to build the model.

3.3 Microsoft Word Dictionary

Microsoft Word dictionary phase appears after a preprocessing phase. In this chapter, a dictionary table is structured depending on Microsoft Word thesaurus to check synonyms of words between students' answers and model answers. When the word of model answers and its synonyms in students' answers have the same meaning, the answer is correct. The probability of knowing correct answers of the students is going to be increased. An example is shown below (Table 1).

3.4 The Collected Data

The data set utilized in this chapter is the same dataset used in [46]. Data set created in Microsoft Excel worksheet as xlsx file, then social books, science and computer in Allu'lu'a modern school were used to get questions. The data are set of questions and answers that combine 40 questions with three classes of answers that leads to a total of 120 answers. Automated essay question grading model is applied in Python Jupyter Notebook because it is easy to implement and is fast to deploy. The data set parameters are as follows:

ID: unique number for each answer.
Score: the score was given by the instructor to student answer.
Answer: it represents answer given by the student.
Question ID: unique number for each question.

A sample of data set containing the questions with ID and ideal answers is shown in Fig. 5.

Table 1 Example of MS Word dictionary

Word	Synonyms
ربط	وصل، شبك، أوثق، قيد
بحث	تحقيق، تفتيش، تنقيب، تقصي
خط	مسار، اتجاه، مسلك
طرق	سبيل، نهج، وسيلة، مسلك، واسطة

Id	Questions	Ideal-answer
1	عرف الانترنت	مجموعة من الأجهزة المتصلة مع بعضها البعض
2	ماذا يقصد بأمن المعلومات	في عملية ابقاء المعلومات تحت سيطرتك الكاملة
3	ماذا تعني بمحركات البحث	هي برامج متخصصة في الشبكة الافراضية تستخدم للبحث عن المعلومات تساعد الباحث للحصول على المعلومات
4	ما هي متطلبات الاتصال بالإنترنت	جهاز حاسوب اشتراك من أحد الشركات المزودة للخدمة ومتصفح انترنت، موديم
5	ماذا يقصد بمزود خدمة الانترنت	شركة تمكن المشترك من الحصول على الانترنت
6	ما هي وظيفة جهاز المودم	جهاز يربط بين الحاسوب وخط الهاتف
7	ما هي الخدمات التي تقدمها الحكومة	تكون الدارة الكهربائية المصباح، الالكترونية، وما فائدتها
8	اذكر خدمات البريد الالكتروني	مشاركة الرسائل والملفات والصور لان
9	علل، احترام الحقوق الملكية الفكرية عند استخدام الانترنت	حقوق النشر ونسخ المواد الموجودة مملوكة لأشخاص آخرين ولا يحق لأحد أن يعيد نشره
10	عرف بروتوكول الانترنت	هي القواعد التي تتحكم في الاتصال وتبادل المعلومات للأجهزة المختلفة المرتبطة بالشبكة
11	عرف الصراع الاجتماعي	عملية اجتماعية تؤدي إلى هدم المجتمع
12	عرف المنصهر	سلك رفيع يوصل في الدارة الكهربائية لحماية الأجهزة الكهربائية من الاحتراق وينصهر عندما يمر فيه تيار قوي.
13	عرف البوصلة	أداة تستخدم لتحديد الاتجاهات وتتركب من مغناطيس صغير يشبه الإبرة ويرتكز على سن مدببة تسمح له بالدوران والاتجاه نحو الشمال

Fig. 5 Sample of data set questions

A sample of data set created for students' answers is shown in Fig. 6, where each question has five answers, three as real answers and two for experimental purpose. Each answer is graded by a human and is given a score. If the answer is fully compatible with the typical answer, the score is 100, but if the answer is partially compatible with the typical answer, the score is 75. When the answer is null, the score is 0. Otherwise, the score is 25.

ID	AnswerId	Score	QuestionID	Answer
1	2	100	1	مجموعه من الحواسيب ترتبط فيما بينها بواسطه خطوط اتصال لها القدره على مشاركه البيانات
2	2	75	1	مجموعه من الحواسيب ترتبط مع بعضها بوسائله خطوط اتصال
3	1	50	1	مجموعه من الحواسيب ترتبط مع بعضها البعض
121	2	100	1	مجموعه من الاجهزة المتصلة مع بعضها البعض
201	2	100	1	مجموعه من الاجهزة المتصلة مع بعضها
161	1	0	1	...
4	2	100	2	هي عمليه ابقاء المعلومات تحت سيطرتك الكامله
5	2	75	2	ابقاء المعلومات تحت سيطره شخص
6	1	50	2	ابقاء المعلومات بحوزتك ومن دون تدخل
122	2	100	2	هي عمليه ابقاء المعلومات تحت سيطرتك الكامله
162	1	0	2	...
7	2	100	3	هي برامج متخصصه في الشبكه الافتراضيه تستخدم للبحث عن المعلومات تساعد الباحث للحصول على المعلومات
8	2	75	3	تساعد الباحث للحصول على المعلومات

Fig. 6 A sample of students' answers

4 Experimental Design and Results

This section presents the result of automated Arabic essay questions grading using naive Bayes (NB) classifier and inner product algorithm similarity measure to score students' answers. The proposed technique carried out all data set to get results. The proposed technique is done by Python Jupyter Notebook (6.0.3) because of the many features that hold. For example, it is easy to use, it supports library for Arabic processing like NLTK library, and it is widely used in machine learning (ML).

4.1 Classification Result

There are different types of text classifier used in this chapter like SVM, NB, and logistic regression. The naive Bayes models have been producing remarkable and successful performance to get high accuracy results. In this chapter, the Gaussian naive Bayes model is used to predict accuracy.

The classifier models are implemented in all data set that contained 200 answers, 120 answers as real data set and 80 answers added for experimental purpose. Table 2 presents accuracy values for implementing classifier models, where naive Bayes (Bernoulli) type with an accuracy value of 0.55 has the lowest accuracy. The highest accuracy value occurs when implementing naive Bayes (Gaussian) type with an accuracy value of 0.9.

Figure 7 presents the graphical description of classification algorithm accuracy results.

4.2 Inner Product Without Using MS Dictionary

Table 3 shows the result when using inner product without MS dictionary.

Table 2 Results of classification accuracy

Model	Accuracy
Linear SVM	0.75
RBF SVM	0.775
Nu SVM	0.75
Gaussian NB	0.9
Multinomial NB	0.575
Bernoulli NB	0.55
Logistic regression	0.75

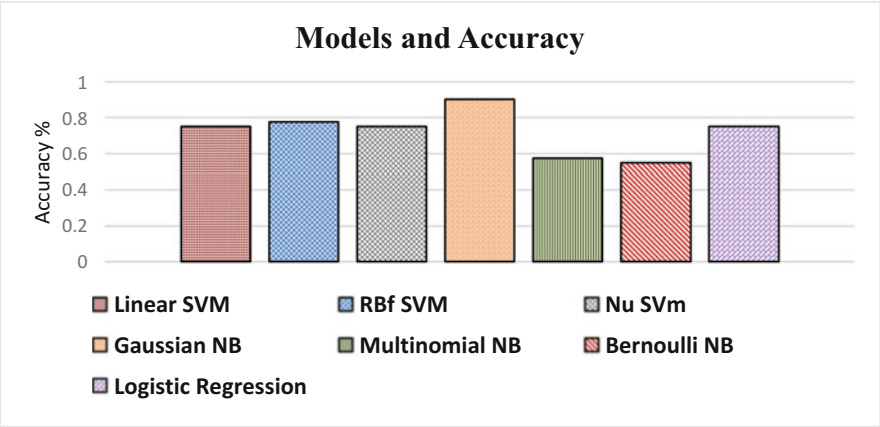

Fig. 7 Classifier accuracy results

Table 3 Result of the proposed model without MS dictionary

Question (ID)	Human score	Inner product result without MS dictionary
1	75	25
2	25	30
8	25	0
10	100	100
11	75	25
14	75	85
21	25	53
23	25	27
29	25	40
35	100	100

Table 4 Result of the proposed model with MS dictionary

Question (ID)	Human score	Inner product result with MS dictionary
1	75	75
2	25	37
8	25	33
10	100	100
11	75	50
14	75	77
21	25	40
23	25	25
29	25	28
35	100	100

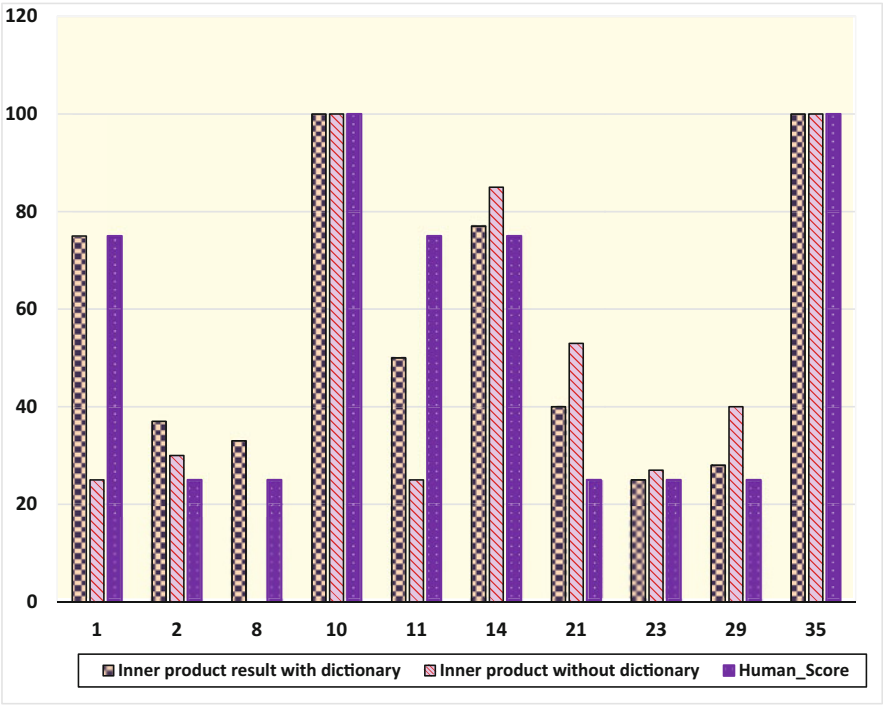

Fig. 8 Rates of inner product similarity with and without dictionary

4.3 Inner Product Using MS Dictionary

Table 4 shows the result when using inner product with MS dictionary.

Figure 8 presents the graphical description comparative result of using inner product with MS dictionary and without MS dictionary.

4.4 Evaluation Results

Evaluation of using Microsoft dictionary in Arabic essay questions grading is performed by comparing human score and automated score using mean absolute error (MAE) along with Pearson correlation result (PCR).

4.4.1 Mean Absolute Error

The MAE evaluates accuracy between human score and the proposed technique. Equation 2 in Sect. 2.4.1 is used to identify the MAE value.

First, MAE for human score and automated Arabic essay questions grading using inner product with dictionary and without using dictionary.

$$\text{Enhancement} = \frac{\text{MAE1} - \text{MAE2}}{\text{MAE1}} * 100\% \tag{4}$$

According to the value of mean absolute error shown in Table 5, the enhanced accuracy is 4.65%.

4.4.2 Pearson Correlation Result

Pearson correlation result calculates the strength between human score and automated technique in order to improve the efficiency of automated essay grading system. Equation 3 is used to calculate Pearson correlation between human score and automated score.

$$\text{Cor}(x, y) = \frac{\sum \text{xy}. \frac{\sum \text{x}. \sum \text{y}}{\text{N}}}{\sqrt{\sum \text{x}^2 - \frac{\sum \text{x}^2}{\text{N}} . \sum \text{Y}^2 - \frac{\sum \text{Y}^2}{\text{N}.}}}$$

where $\boldsymbol{x}$ is the human score (dependent variable), $\boldsymbol{y}$ is the automated score (independent variable), and $\boldsymbol{N}$ is the number of question test

(independent variable).

Table 5 Mean absolute error for the proposed technique with and without MS dictionary

	Mean absolute error (MAE)
Inner product with MS dictionary	0.041
Inner product without MS dictionary	0.043

Table 6 Pearson correlation results for inner product with MS dictionary and without MS dictionary

	Inner product with MS dictionary	Inner product without MS dictionary
Pearson correlation result	0.8250	0.8193

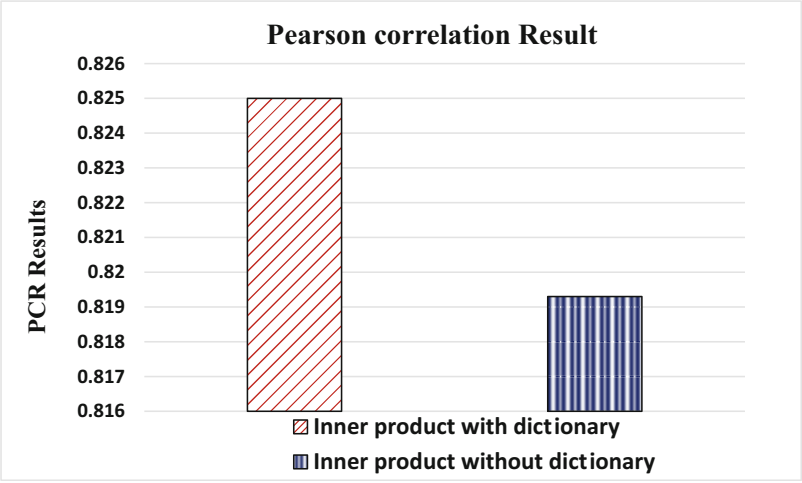

Fig. 9 Pearson correlation

The valid result for Pearson correlation is between −1 and +1. If the correlation result is between 0.5 and 1, that means high positive correlation between x variable and y variable. If the correlation result is between 0 and 0.49, that means low positive correlation. If the correlation result is between −0.5 and 0, that means negative correlation. If the correlation result is between −0.49 and −1, that means strong negative correlation.

As shown in Table 6, the value of Pearson correlation is between 0.5 and 1, which means that this is a strong positive correlation and high *X* variable scores go with high *Y* variable scores and vice versa.

Figure 9 presents the graphical description of the PCR for automated essay using MS dictionary and without using MS dictionary. The Pearson correlation result between human score and the proposed technique while using inner product similarity with MS dictionary is larger than that of inner product without using MS dictionary, which means that the proposed technique will improve Arabic essay questions technique when utilizing MS dictionary.

5 Conclusions and Future Work

This chapter represented Arabic essay question grading techniques with the use of naive Bayes classifier to get classification accuracy and inner product similarity in order to get the score of students' answers. The aim of this work is to enhance automated essay questions so that they can match human score by adding Microsoft Word dictionary. Because getting more choices for students' answers leads to prove the efficiency of the proposed technique. All of this is performed on various Arabic data sets that contain 40 questions with 120 answers.

Many experiments were made by implementing automated essay scoring system on students' answers and then comparing the results.

The results showed improvement on automated system even when using inner product similarity with Microsoft Word dictionary. The results are better in terms of accuracy compared with automated system when using inner product without Microsoft Word dictionary. This was proved by calculating the quality metrics mean absolute error (MAE) along with Pearson correlation result (PCR), where MAE for inner product with dictionary is 0.041 and PCR is 0.825. While comparing with inner product but without MS dictionary, the value of MAE is 0.043 and PCR is 0.819.

Future suggestions and recommendations for this chapter are to implement the proposed technique on different types and larger data set, considering different classification theories and similarity algorithms on same data set. In addition, the ability to evolve voice recognition on automated Arabic essay questions scoring instead of the use of written essay questions.

References

1. W.H. Gomaa, A.A. Fahmy, Arabic short answer scoring with effective feedback for students. Int. J. Comp. Appl. **86**(2), 35–41 (2014)
2. A. Shehab, M. Faroun, M. Rashed, An automatic Arabic essay grading system based on text similarity algorithms. Int. J. Adv. Comput. Sci. Appl. **9**(3) (2018)
3. D. Hutchison, *Automated Essay Scoring Systems*, in *Handbook of Research on New Media Literacy at the K-12 Level: Issues and Challenges*, (IGI Global, Hershey, 2009), pp. 777–793
4. M.M. Refaat et al., Automated assessment of students Arabic free-text answers. IJICIS **12**(1), 213–222 (2012)
5. K.M.O. Nahar, I.M. Alsmadi, The automatic grading for online exams in Arabic with essay questions using statistical and computational linguistics techniques. MASAUM J. Comput. **1**(2), 215–220 (2009)
6. D.S.V. Madala et al., An empirical analysis of machine learning models for automated essay grading. PeerJ Preprints (2018). https://doi.org/10.7287/peerj.preprints.3518v1
7. L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J. Supercomput. **73**(11), 4773–4795 (2017)
8. L.M. Abualigah et al., Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Expert Syst. Appl. **84**, 24–36 (2017)

9. Z. yong, L. Youwen, X. Shixiong, An improved KNN text classification Algorithm based on clustering. J. Comput. **4**(3), 230–237 (2009)
10. Y.M. Fanny, F. Tanzil, *A comparison of text classification methods K-NN, Naive Bayes and support vector machine for news classification*. J Pengembangan IT **3**(2) (2018)
11. S. Jimenez, C. Becerra, G. Alexander, Soft Cardinality: A Parameterized Similarity Function for Text Comparison, in *First Joint Conference on Lexical and Computational Semantics (*SEM)*, (2012), pp. 449–453
12. S.-H. Cha, Comprehensive survey on distance/similarity measures between probability density functions. Int. J. Math. Methods Appl. Sci. **1**(4), 300–307 (2007)
13. L. Abualigah et al., The arithmetic optimization algorithm. Comput. Methods Appl. Mech. Eng. **376**, 113609 (2021)
14. A.H. Mohammad, T. Alwada'n, O. Al-Momani, Arabic text categorization using support vector machine, Naive Bayes and neural network. GSTF J. Comput. (JOC) **5**(1), 108–115 (2016)
15. V. Korde, C.N. Mahender, Text classification and classifiers: a survey. Int. J. Artif. Intel. Appl. (IJAIA) **3**(2) (2012)
16. L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm. J. Comput. Sci. **25**, 456–466 (2018)
17. L.M. Abualigah, A.T. Khader, E.S. Hanandeh, Hybrid clustering analysis using improved krill herd algorithm. Appl. Intell. **48**(11), 4047–4071 (2018)
18. L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis. Eng. Appl. Artif. Intell. **73**, 111–125 (2018)
19. L.M. Abualigah et al., A novel hybridization strategy for krill herd algorithm applied to clustering techniques. Appl. Soft Comput. **60**, 423–435 (2017)
20. G. Kanaan et al., A comparison of text-classification techniques applied to Arabic text. J. Am. Soc. Inf. Sci. Technol. **60**(9), 1836–1844 (2009)
21. S. Alsaleem, Automated Arabic text categorization using SVM and NB. Int. Arab J. e-Technol. **2**(2), 124–128 (2011)
22. Al-Shargabi, B., W. AL-Romimah, And F. Olayah, A Comparative Study for Arabic Text Classification Algorithms Based on Stop Words Elimination. ACM Digital Library, New York, NY 2011
23. S. Warmerdam, *Features for Instrument Recognition in Polyphonic Mixes* (Delft University of Technology, Delft, 2017)
24. S. Xu, Bayesian Naıve Bayes classifiers to text classification. J. Inf. Sci. **44**(1), 48–59 (2018)
25. *Support Vector Machine Algorithm*. www.javatpoint.com. (2018)
26. L.M.Q. Abualigah, *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering* (Springer, Switzerland, 2019)
27. L.M.Q. Abualigah, E.S. Hanandeh, Applying genetic algorithms to information retrieval using vector space model. Int. J. Comput. Sci. Eng. Appl. **5**(1), 19 (2015)
28. M.K. Vijaymeena, K. Kavitha, A survey on similarity measures in text mining. Mach. Learn. Appl. **3**(1) (2016)
29. W.H. Gomaa, A.A. Fahmy, A survey of text similarity approaches. Int. J. Comput. Appl. **68**(13), 13–18 (2013)
30. S.A.A. Awaida, B. Al-Shargabi, T. Al-Rousan, Automated Arabic essay grading system based on f-score and Arabic wordnet. Jordan. J. Comput. Inform. Technol. (JJCIT) **5**(3), 1 (2019)
31. M.F. Al-Jouie, A.M. Azmi, Automated evaluation of school children essays in Arabic. Proc. Comput. Sci. **117**, 19–22 (2017)
32. H. Rababah, T.A. Al-Taani, *An Automated Scoring Approach for Arabic Short Answers Essay Questions*, in *2017 8th International Conference on Information Technology (ICIT)*, (2017), pp. 697–702
33. E.F. Al-Shalabi, *An automated system for essay scoring of online exams in Arabic based on stemming techniques and Levenshtein edit operations*. Int. J. Comput. Sci. **13**(5) (2016)

34. R. Mezher, N. Omar, A hybrid method of syntactic feature and latent semantic analysis for automatic Arabic essay scoring. J. Appl. Sci. **16**(5), 209–215 (2016)
35. W.H. Gomaa, A.A. Fahmy, Short answer grading using string similarity and corpus-based similarity. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **3**, 11 (2012)
36. A. Khan et al., A review of machine learning algorithms for text-documents classification. J. Adv. Inform. Technol. **1**(1), 4–20 (2010)
37. A. Ayedh et al., *The effect of preprocessing on Arabic document categorization*. *Algorithms* **9**(2), 27 (2016)
38. A. Patra, D. Singh, *A survey report on text classification with different term weighing methods and comparison between classification algorithms*. Int. J. Comput. Appl. **75**(7) (2013)
39. L. Abualigah et al., Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis. Algorithms **13**(12), 345 (2020)
40. D. Sharma, Stemming algorithms: a comparative study and their analysis. Int. J. Appl. Inform. Syst. (IJAIS) **4**(3) (2012)
41. L. Abualigah et al., Advances in meta-heuristic optimization algorithms in big data text clustering. Electronics **10**(2), 101 (2021)
42. M.G. Syarief et al., Improving Arabic Stemmer: ISRI Stemmer, in *In 2019 IEEE 5th International Conference on Wireless and Telematics (ICWT)*, (IEEE, Yogyakarta, Indonesia, 2019)
43. M.A. Otair, *Comparative analysis of Arabic stemming algorithms*. Int. J. Manag. Inform. Technol. (IJMIT) **5**(2) (2013)
44. K. Taghva, R. Elkhoury, J. Coombs, Arabic stemming without a root dictionary. ITCC **2005**(1), 152–157 (2005)
45. S.L. Ting, W.H. Ip, A.H.C. Tsang, *Is Naïve Bayes a good classifier for document classification?* Int. J. Soft. Eng. Appl. **5**(3) (2011)
46. S.E.O. Al-awaida, *Automated Arabic Essay Grading System based on Support Vector Machine and Text Similarity Algorithm* (Middle East University, Amman, Jordan, 2019)

Optimal Fractal Feature Selection and Estimation for Speech Recognition Under Mismatched Conditions

Puneet Bawa, Virender Kadyan, Archana Mantri, and Vaibhav Kumar

1 Introduction

The discrepancies between human speech and the more conventional types of machine feedback are the basic differences in the use of speech as a part of a computer simulation. Although these simulation systems are usually programmed with an ultimate objective of producing an acceptable range of defined output. It is entirely dependent on the near-precise transformation of human speech signal [1, 2]. However, the human voice is unique in such a way that even the same words can mean different when spoken in different ways or under varying environmental conditions. In the pandemic situation, children are now learning how to use computers in schools as well as in their homes at their earlier stages [3]. With more and more voice-based applications being designed, it is vital that children are able to use advents of such technologies [4, 5]. Thus, the capacity of speech recognition systems to manage the proper pronunciation corresponding to each vowel and word thus becomes considerably important for improved productivity. Moreover, children typically face issues with generating and pronouncing vowels in the case of tonal

P. Bawa · V. Kumar
Centre of Excellence for Speech and Multimodal Laboratory, Chitkara University Institute of Engineering & Technology, Chitkara University, Chandigarh, Punjab, India
e-mail: puneet.bawa@chitkara.edu.in

V. Kadyan (✉)
Speech and Language Research Centre (SLRC), School of Computer Science, University of Petroleum & Energy Studies, Dehradun, Uttarakhand, India
e-mail: vkadyan@ddn.upes.ac.in

A. Mantri
Chitkara University Institute of Engineering & Technology, Chitkara University, Chandigarh, Punjab, India
e-mail: archana.mantri@chitkara.edu.in

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_3

languages [6]. Thus, an effective recognition of vowels under such scenarios is important since they immediately impede intelligibility and negligence may inflict severe effects on consonant productions. The words spoken by children likewise coincide high fundamental frequencies in comparison to adult speech where the constant shift in vocal apparatus is evident during the growth of children. Also, the collection and processing of children spoken data are often a tedious, repetitive, and resource-consuming task such that multiple endeavors for the adaptation of adult speech onto children have been studied and experimented. Also, emotions can be extracted from an input signal using a set of convolutional layers and deep speech layers. The model can help to achieve state-of-the-art results for detecting joy and sorrow [7].

Thus, various efforts for the optimal extraction and selection of speech feature vectors have been attempted by the researchers with a much broader scope of success. These efforts for ideal identification of voice can be classified using the two types of methods: (a) template matching [8] and (b) feature analysis. The template matching approach is the simplest technique, which is a close correspondence with conventional keyboard inputs relying on the basic conditional assertions to fit the actual input correctly. This method is therefore only acceptable for designing speaker-dependent ASR systems where the input digitized voice sample recognizes only the previously modelled contexts [8, 9]. On the other hand, the much reliable methodology of feature analysis is used as a general method for designing speaker-independent ASR systems. Unlike template matching, this method processes an input speech signal based on Fourier transformation, and then feature-like correlations are located between the existing digitized sample and the predicted inputs [10, 11]. Likewise, the nonlinear behavior of signals can be seen in almost every digital communication system, ranging from the use of wireless networks to cellular to satellite systems [12, 13]. As a result, several nonlinear operations and techniques for the appropriate extraction of additional features corresponding to the input speech signal have been implemented. Among these methods, the latest streamlined techniques utilized are based on the fractal analysis for precise estimation of nonlinear signal dimensionality [14]. In these methods, the corresponding analysis is focused on fractal classification by exploring the time series mainly in the case of biomedical signals including electrocardiogram, normal speech signals, and electroencephalogram [15–18].

In this research, an effort has been made to address such vowel pronunciation difficulties present in children' speech using the optimal selection of three fractal feature dimensional analysis techniques—Katz FD, Higuchi FD, and Petrosian FD. As in the case of Punjabi children speech recognition system, there is presence of a greater fractal indicator suggesting a higher irregularity throughout the time series. These anomalies might probably correspond to higher FD values and increased discrepancies between training and testing utterance. The motive of the experiments is the successful development of methods and algorithms, which is able to substantially reduce the calculational complexities and lead to the increased effectiveness by employing the above factors of existing channel nonlinearity. Thus, an enhanced Punjabi children system for adequate recognition of a more

complicated and high-frequency speech signal that is less stationary has been proposed. Finally, the increased performance has been noted after the evaluation of individual fractal characteristics utilizing the hybrid acoustic classifier DNN-HMM.

The remainder of this chapter is aligned as follows: Sect. 2 discusses the literature review undertaken by the researchers to demonstrate the need for appropriate algorithms to eliminate anomalies from an input speech signal. Further, Sect. 3 discusses the theoretical background of the techniques which are being utilized for better analysis of the proposed system and later its prediction for learning of an input signal are presented in Sect. 4. The experimental results are described in Sect. 5. Finally, the study is concluded in Sect. 6.

2 Related Work

Initial study for fractal features was directed in 1975 [19, 20], as a part of present-day numerical hypothesis. The fractal dimension is the leading parameter in fractal theory. It quantitatively depicts the unpredictability of the fractal set. As a sort of time series, the correspondence tweak sign can be successfully depicted by the fractal measurement. The basic one-dimensional fractal dimensions are Hausdorff–Besicovitch dimension (HBD), Higuchi dimension, Katz dimension, Petrosian dimension, and Sevcik fractal dimension. HBD [21] was the basic fractal dimension that was considered to limit the radio wire size, holding a high radiation effectiveness [22]. This confirmed the importance of HBD and, it also showed that it has a high computational unpredictability, which made it hard to acknowledge the speaker based information. To measure the fractal dimension of waveforms, the Sevcik fractal dimensions were proposed [23]. This approach may be used to easily test HBD waveform measurements and calculate the various facets of waveforms and arbitrariness. Petrosian suggested a fast procedure for estimating a finite sequence fractal dimension [24], converting the data into a binary sequence, before estimating the time-series fractal dimension. Tricot thought about the assessment precision conveyed by Katz's and Higuchi's techniques on four manufactured fractal waveforms. The outcomes showed that Katz's strategy perpetually under-assessed the genuine measurement; however Higuchi's technique was more precise in assessing the fractal features. In addition, Ezz-Eldin et al. [25] proposed the concatenated output of its proposed hybrid networks. It also tried to fed it into a softmax layer. It also tried to generate a probability distribution through categorical language recognition categories. It can be considered as a genuine measurement on the basis of finding and the accuracy against its input features [26].

In [27], fractal demonstration and estimation were explored in order to implement fractal functions in engineering. In [14], several fractals have been examined in various noise propagation systems for a wireless communication signal. The simulation results showed that the FSK (frequency shift keying) function of different noise distributions could actually be extracted by various fractals. The most skilled

tool for selecting fractal calculation highlights for organizing connectivity modulation signs was aimed at an effective observational research in [28]. In [29], an Automatic Speech Recognition (ASR) framework was presented with very few training samples. It employed fractal features from an observed speech waveform. In order to enhance the performance of two different ASR systems, it was combined with a traditional MFCCs. Further, in [30], the researchers proposed different combinations of feature extraction techniques: MFCC, RASTA-PLP, and PLP. The research paved the way to overcome the challenge of performance degradation due to inconsistency between training and testing phases. In [31], researchers experimented with different sound waves such as mixture of sounds, noise, etc., and came up with a system to iteratively remove noise and reverberation effect and enhance the speech under noisy environment. Theoretically, it is already known that there is a mismatch between children and adult speech. In [32], authors explored the incorporation of mismatched training data to achieve a better acoustic model and improve performance in the face of limited training data, as well as training data augmentation using noise. Manjutha et al. [33] also experimented with selecting optimal features from MFCC using particle swarm optimization and synergistic fibroblast optimization for Tamil dataset and fed the features into machine learning algorithms such as naive Bayes and support vector machines to yield a better accuracy. In [34], authors performed comparative study between baseline Mel frequency cepstral coefficient features and gammatone frequency cepstral coefficient features with DNN-HMM modeling to resolve the problem of overfitting and set up a pipeline for Punjabi ASR systems. In [35], authors explored various feature extraction techniques with N-gram language model and monophone and triphone acoustic modeling and found that MFCC feature extraction gave better accuracy as compared to others. In Bawa and Kadyan (2021), researchers handled acoustic and phonetic variations among adult and children speeches using gender-based in-domain training data augmentation, and later acoustic variability among speakers in training and testing sets was normalized using vocal tract length normalization. The results yielded significant system performance over different environmental conditions, i.e., clean or noisy [36]. In [37], authors empirically determined optimal feature for improving accuracy of ASR system and found out that features like pitch for tonal languages such as Punjabi and probability of voicing can help elevate the performance of ASR systems. In [38], authors explored end-to-end attention-based network for language identification systems. The proposed method proved to outperform state-of-the-art models.

3 Feature Extraction

3.1 Katz Fractal Features

A speech signal (a_i, b_i) of duration D with W_L as length of waveform and $d_{\max}$ as the measure of maximum distance between initial point (a_i, b_i) and other points can be used to obtain Katz fractal feature (K_z):

$$K_z = \frac{\log(D)}{\log(D) + \log\left(\frac{d_{\max}}{W_L}\right)} \tag{1}$$

$$W_L = \sum_{i=0}^{D-2} \sqrt{(b_{i+1} - b_i)^2 + (a_{i+1} - a_i)^2} \tag{2}$$

$$d_{\max} = \max\left(\sqrt{(b_i - b_1)^2 + (a_i - a_1)^2}\right) \tag{3}$$

3.2 Higuchi Fractal Dimension

Higuchi's method is recursive in nature like the Katz method. It has proven to be handy over the years especially for waveforms and signals.

Considering a speech signal as $a(1)$, $a(2)$, ... , $a(N)$ having k as the interval between two adjacent time series, $m = 1, 2, \ldots, k$ as the initial time value of the sequence, the recomputed time sequence is shown as

$$a_m^k = \left\{ a(m), a(m+k), a(m+2k), \ldots, a\left(m + \left[\frac{N-m}{k}\right].k\right)\right\} \tag{4}$$

For every a_m^k, the average length $AL_m(k)$ is shown as

$$AL_m(k) = \frac{\sum_{i=1}^{\left[(N-M)/k\right]} |\, a(m+ik) - a(m+(i-1).k)|.(N-1)}{\left[(N-M)/k\right].k} \tag{5}$$

Hence, the total average length of the discrete speech signal can be represented as

$$AL_T(k) = \sum_{n=1}^{k} AL_m(k) \tag{6}$$

By the logarithmic transformation, the Higuchi fractal dimension H can be determined as

$$\ln(AL_T(k)) \propto H.\ln\left(\frac{1}{k}\right) \tag{7}$$

3.3 Petrosian Fractal Dimension

Primarily, the speech time series can be represented as $a(1), a(2), \ldots\ldots, a(N)$. Given that an input speech signal is composed of series of points $b_1, b_2, b_3, \ldots\ldots, b_N$. A binary sequence $c_i$needs to be generated from this as

$$c_i = \begin{cases} 1, & a_i > \text{mean}(b) \\ -1, & a_i \leq \text{mean}(b) \end{cases}, i = 1, 2, \ldots., N \tag{8}$$

Then, the total number of adjacent symbol changes in the sequence can be computed as

$$\text{TN}_\Delta = \sum_{i=1}^{N-2} \left| \frac{c_{i+1} - c_i}{2} \right| \tag{9}$$

The Petrosian fractal dimension can be determined as

$$P = \frac{\log_{10} N}{\log_{10} N + \log_{10}\left(\frac{N}{N+0.4\text{TN}_\Delta}\right)} \tag{10}$$

4 Proposed System Architecture

The experiments were implemented on 32 children speech speakers and 21 adult speech speakers being sampled at a frequency of 16 kHz. Audios for the database have been recorded in both open and closed environments with and without the use of a microphone. The initial method for speech recognition is the characterization of complex representations and patterns in the input speech signal by the extraction of relevant features for which the commonly used front-end MFCC extraction technique has been utilized with 25 ms frame size and 10 ms frame shift. Frequency bands are determined on the basis of the frequency domain of pre-processed signal $y(t)$ in such a way that the transition between the Mel Scale formulated as function of *Mel* and the frequency $f(y(t))$ is carried out using

$$Mel\ (\ f(y(t))) = 2595\ \log_{10}\left(1 + \frac{f(y(t))}{700}\right) \tag{11}$$

Likewise, the dual stages have been employed for enhanced extraction of features corresponding to the values of fractal dimension with an objective of eliminating the inconsistencies existing in an input speech signal. Initially, the frame-level features

corresponding to an input speech signal are extracted, and the classical frame-by-frame analysis of all utterances in corpora was carried out. Thus, correspondingly the exponential values of fractal dimension corresponding to Katz(kfd_{feats}), Higuchi (hfd_{feats}) and Petrosian(pfd_{feats}) are evaluated in every frame for obtaining adequate time series for some utterances of adult speech as detailed in Fig. 1a and children speech in Fig. 1b.

Algorithm 1 Procedure for the optimal selection of features for processing under mismatched conditions

Step 1: Initialize the adult speech corpus as $\text{adult}_{\text{corp}}$ and children speech corpus as $\text{child}_{\text{corp}}$.

Step 2: Divide the training set of $\text{adult}_{\text{corp}}$ as 80 – 20%and $\text{child}_{\text{corp}}$ as 80 – 20%and merge them to produce concatenated corpora of sets.

$\text{finalTrain}_{\text{set}}$ = 80 % of ($\text{adult}_{\text{corp}}$) + 80 % of ($\text{child}_{\text{corp}}$)

$\text{finalTest}_{\text{set}}$ = 20 % of ($\text{adult}_{\text{corp}}$) + 20 % of ($\text{child}_{\text{corp}}$)

Step 3: Extract the corresponding MFCC features for the produced training and testing set using Eq. (11):

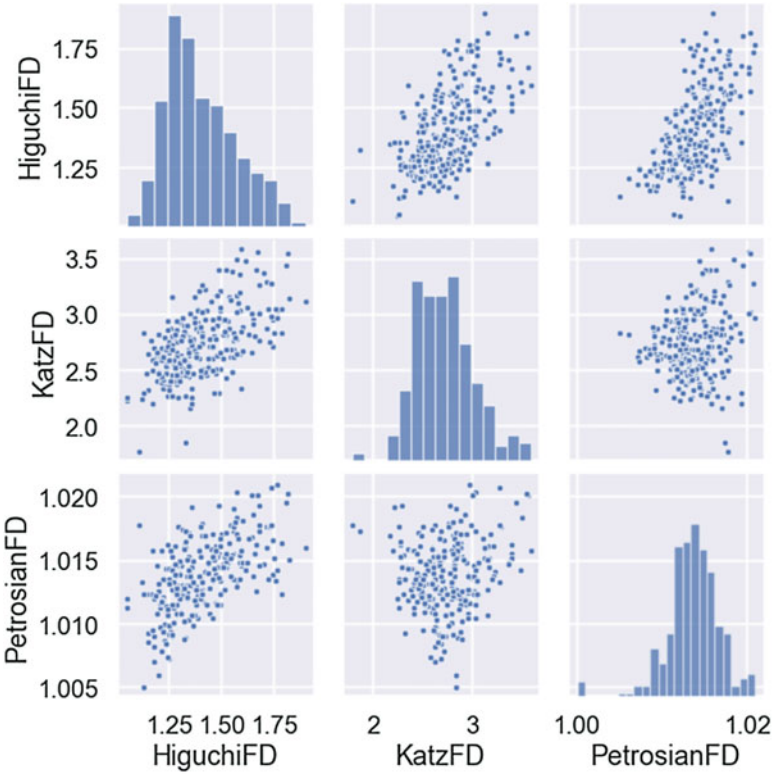

Fig. 1 (**a**) Pair plot representation of fractal dimension-based features for adult speech corpora. (**b**) Pair plot representation of fractal dimension-based features for children speech corpora

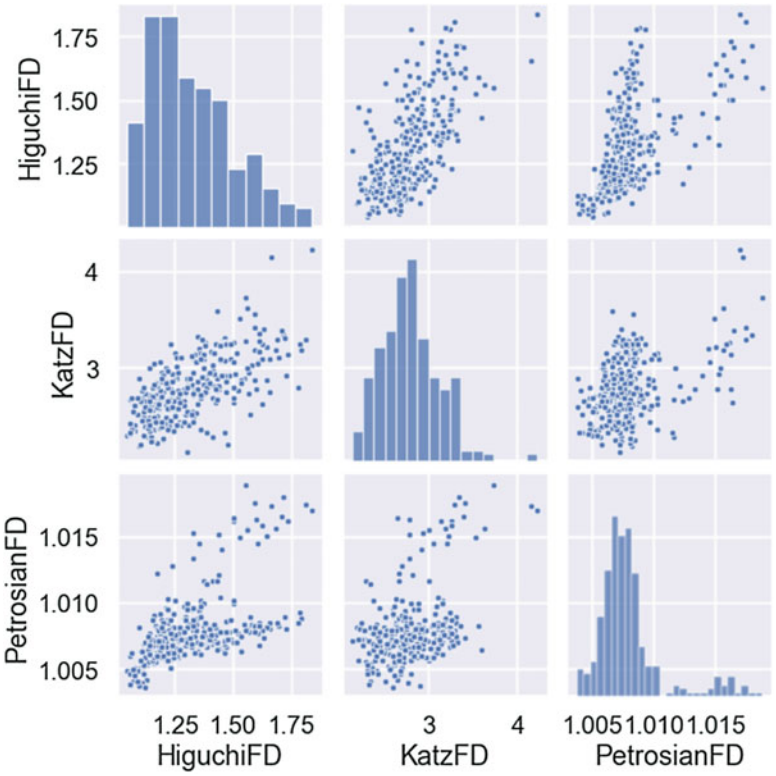

Fig. 1 (continued)

$\text{mfccTrain}_{\text{feats}} = \text{mfcc}(\text{finalTrain}_{\text{set}})$
$\text{mfccTest}_{\text{feats}} = \text{mfcc}(\text{finalTest}_{\text{set}})$
Step 4: Extract the individual fractal features as over the given signal:

Step 4.1: Extract Katz fractal dimension features using Eq. (1):
$\text{kfdTrain}_{\text{feats}} = \text{kfd}(\text{finalTrain}_{\text{set}})$
$\text{kfdTest}_{\text{feats}} = \text{kfd}(\text{finalTest}_{\text{set}})$
Step 4.2: Extract Higuchi fractal dimension features using Eq. (4):
$\text{hfdTrain}_{\text{feats}} = \text{hfd}(\text{finalTrain}_{\text{set}})$
$\text{hfdTest}_{\text{feats}} = \text{hfd}(\text{finalTest}_{\text{set}})$
Step 4.3: Extract Petrosian fractal dimension features using Eqs. (8 and 9):
$\text{pfdTrain}_{\text{feats}} = \text{pfd}(\text{finalTrain}_{\text{set}})$
$\text{pfdTest}_{\text{feats}} = \text{pfd}(\text{finalTest}_{\text{set}})$

Step 5: Concatenate the features for both training and testing dataset:

Step 5.1: Concatenate the Katz fractal dimension as extracted in **step 4.1** with MFCC features as extracted features in **step 3** with representation as first set of fractal features ff_1:

train _ $ff_1 \leftarrow [mfccTrain_{feats} + kfdTrain_{feats}]$
test _ $ff_1 \leftarrow [mfccTest_{feats} + kfdTest_{feats}]$

Step 5.2: Concatenate the Higuchi fractal dimension as extracted in **step 4.2** with MFCC features as extracted features in **step 3** with representation as second set of fractal features ff_2:

train _ $ff_2 \leftarrow [mfccTrain_{feats} + hfdTrain_{feats}]$
test _ $ff_2 \leftarrow [mfccTest_{feats} + hfdTest_{feats}]$

Step 5.3: Concatenate the Petrosian fractal dimension as extracted in **step 4.3** with MFCC features as extracted features in **step 3** with representation as second set of fractal features ff_3:

train _ $ff_3 \leftarrow [mfccTrain_{feats} + pfdTrain_{feats}]$
test _ $ff_3 \leftarrow [mfccTest_{feats} + pfdTest_{feats}]$

Further, the decision is to take concatenate original MFCC characteristics and independently extracted characteristics in the second stage after extraction of these features as shown in Fig. 2 and detailed in Algorithm 1. These resilient characteristics are concatenated with the purpose of identifying appropriate features corresponding to children voice, as the related FD qualities are extremely varied and reflect individual speaker's status characteristics. These rendered features FF1 ($mfcc_{feats} + kfd_{feats}$), FF2($mfcc_{feats} + hfd_{feats}$), and FF3($mfcc_{feats} + pfd_{feats}$) are being processed individually on mixed dataset resulting into 42 features after being normalized using the process of cepstral mean and variance normalization (CMVN) [39].

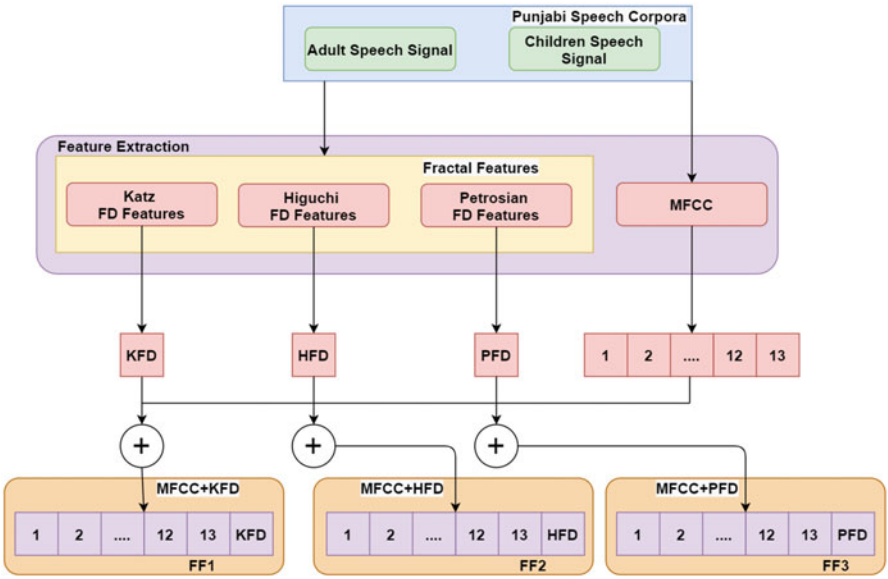

Fig. 2 Block diagram summarizing the steps for extraction of concatenated features

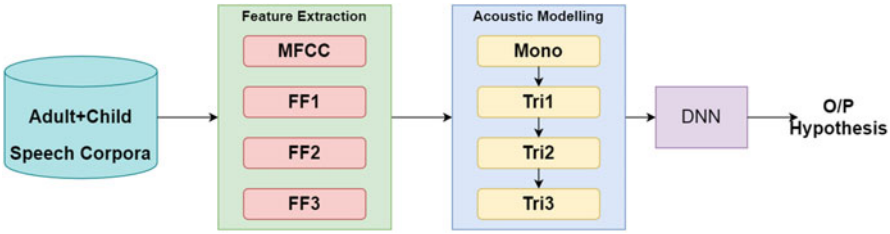

Fig. 3 Block diagram of the proposed architecture for optimal selection of heterogeneous features required for adequate acoustic modeling

Table 1 WER (%) for various heterogeneous feature extraction techniques under mismatched conditions

	MFCC	FF1	FF2	FF3
Child	15.43	15.19	15.11	15.31
Adult-child	14.27	13.86	13.65	13.88

The individual features are fed into the DNN-HMM hybrid classifier following the extraction process. During acoustic modeling, monophonic (mono), delta (tri1), and delta-delta (tri2)-based trainings are used to better process the features achieved. This results into 117 feature dimensions as in the case of MFCC feature extraction and 126 in the case of FF1, FF2, and FF3 as detailed in Fig. 3. For the neural network, these high-dimensional speech features are difficult to interpret and are thus converted into a 40-dimensional process using MLLT (maximum likelihood linear transform). Finally, the results using Kaldi-based DNN [40] are evaluated with the WER (%) and RI efficiency metrics (%).

5 Performance Evaluation

The experiments on the two types of datasets, (a) children corpora and (b) adult-children pooled corpora, were conducted for testing the adequateness of fractal dimension-based feature sets with detailed results as shown in Table 1. Initially, the baseline system was developed showcasing better performance of these heterogeneous features involving FF1, FF2, and FF3 with respective relative improvement of 1.55%, 2.07%, and 0.7% in comparison to MFCC. As the children dataset is redundant, the experiments employing the augmentation strategy with adaptation of adult data onto children data have been implemented. A high heterogeneity of the chosen subsets for optimal selection of various combinations of features has been achieved concerning the use of limited acoustic functionalities. Nevertheless, the pooled feature utilizing HFD (FF2) has performed well over other fractal dimension-based features with relative improvement of 1.52% and 1.66% in comparison to FF1 and FF2. This analysis has finally led to a steady development by expansion of lower feature sets representing an enhanced children ASR system with overall relative improvement of 11.54% in comparison to the baseline system.

6 Conclusion

Given the nonlinear and fluctuating essence of the words being spoken by children, the fractal-dimension-based features have been applied. The feature selection has helped in easier classification for generating adequate feature vectors instead of employing the full set of features. Likewise, the selection employing Higuchi fractal dimension has provided relative improvement of 2.07% and 4.34% under normal and mismatched conditions, respectively. The obtained selection employing the adaptive data augmentation has enabled us in the exploration of new features set under mismatched conditions and thereby has led to overall improvement of children ASR system by 11.54%. These strategies for extraction of adequate fractal dimensions can be regarded as extremely selective for building children speech recognition system. The in-depth exploration of heterogeneous fractal measurements with definitions that rely on prosodic characteristics [41] will in the near future widen an intriguing trail of this research and make it linked specially for adequate recognition of children speech.

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. L.R. Rabiner, Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, in *Readings in Speech Recognition*, (IEEE, 1990), p. 267
2. W. Zhang, Y. Liu, X. Wang, X. Tian, The dynamic and task-dependent representational transformation between the motor and sensory systems during speech production. Cogn. Neurosci. **11**(4), 194–204 (2020). https://doi.org/10.1080/17588928.2020.1792868
3. J. Wolfe, J. Smith, S. Neumann, S. Miller, E.C. Schafer, A.L. Birath, et al., Optimizing communication in schools and other settings during COVID-19. Hear. J. **73**(9), 40–42 (2020). https://doi.org/10.1097/01.HJ.0000717184.65906.b9
4. D. Giuliani, M. Gerosa, Investigating recognition of children's speech, in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP'03)*, vol. 2, (IEEE, 2003), p. II-137. https://doi.org/10.1109/ICASSP.2003.1202313
5. M. Russell, C. Brown, A. Skilling, R. Series, J. Wallace, B. Bonham, P. Barker, Applications of automatic speech recognition to speech and language development in young children, in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 1, (IEEE, 1996), pp. 176–179. https://doi.org/10.1109/ICSLP.1996.607069
6. J.L. Wu, H.M. Yang, Y.H. Lin, Q.J. Fu, Effects of computer-assisted speech training on Mandarin-speaking hearing-impaired children. Audiol. Neurotol. **12**(5), 307–312 (2007)
7. J. Oliveira, I. Praça, On the usage of pre-trained speech recognition deep layers to detect emotions. IEEE Access **9**, 9699–9705 (2021)
8. S. Dey, P. Motlicek, S. Madikeri, M. Ferras, Template-matching for text-dependent speaker verification. Speech Comm. **88**, 96–105 (2017). https://doi.org/10.1016/j.specom.2017.01.009
9. A. Arora, V. Kadyan, A. Singh, Effect of Tonal Features on Various Dialectal Variations of Punjabi Language, in *Advances in Signal Processing and Communication: Select Proceedings of ICSC 2018*, ed. by B. S. Rawat, A. Trivedi, S. Manhas, V. Karwal, (Springer, New York, 2018), pp. 467–472

10. N. Bassan, V. Kadyan, An Experimental Study of Continuous Automatic Speech Recognition System Using MFCC with Reference to Punjabi, in *Recent Findings in Intelligent Computing Techniques: Proceedings of the 5th ICACNI 2017*, vol. 707, (Springer Nature, Singapore, 2018), p. 267. https://doi.org/10.1007/978-981-10-8639-7_288
11. Y. Kumar, N. Singh, M. Kumar, A. Singh, AutoSSR: An efficient approach for automatic spontaneous speech recognition model for the Punjabi language. Soft Comput.., Springer (2020). https://doi.org/10.1007/s00500-020-05248-1
12. A. Chern, Y.H. Lai, Y.P. Chang, Y. Tsao, R.Y. Chang, H.W. Chang, A smartphone-based multi-functional hearing assistive system to facilitate speech recognition in the classroom. IEEE Access **5**, 10339–10351 (2017). https://doi.org/10.1109/ACCESS.2017.27114899
13. Z. Zhang, J. Geiger, J. Pohjalainen, A.E.D. Mousa, W. Jin, B. Schuller, Deep learning for environmentally robust speech recognition: An overview of recent developments. ACM Trans. Intel. Syst. Technol. (TIST) **9**(5), 1–28 (2018). https://doi.org/10.1145/3178115
14. H. Wang, J. Li, L. Guo, Z. Dou, Y. Lin, R. Zhou, Fractal complexity-based feature extraction algorithm of communication signals. Fractals **25**(04), 1740008 (2017). https://doi.org/10.1142/S0218348X17400084
15. M. Dalal, M. Tanveer, R.B. Pachori, Automated Identification System for Focal EEG Signals Using Fractal Dimension of FAWT-based Sub-bands Signals, in *Machine Intelligence and Signal Analysis*, (Springer, Singapore, 2019), pp. 583–596. https://doi.org/10.1007/978-981-13-0923-6_50
16. J. Kaur, A. Singh, V. Kadyan, Automatic Speech Recognition System for Tonal Languages: State-of-the-Art Survey, in *Archives of Computational Methods in Engineering*, (Springer, 2020). https://doi.org/10.1007/s11831-020-09414-4
17. A. Korolj, H.T. Wu, M. Radisic, A healthy dose of chaos: Using fractal frameworks for engineering higher-fidelity biomedical systems. Biomaterials **219**, 119363 (2019). https://doi.org/10.1016/j.biomaterials.2019.119363
18. A. Singh, V. Kadyan, M. Kumar, N. Bassan, ASRoIL: a comprehensive survey for automatic speech recognition of Indian languages. Artif. Intel. Rev.., Springer **53**, 3673–3704 (2019)
19. J.P.A. Sanchez, O.C. Alegria, M.V. Rodriguez, J.A.L.C. Abeyro, J.R.M. Almaraz, A.D. Gonzalez, Detection of ULF geomagnetic anomalies associated to seismic activity using EMD method and fractal dimension theory. IEEE Lat. Am. Trans. **15**(2), 197–205 (2017). https://doi.org/10.1109/TLA.2017.7854612
20. Y.D. Zhang, X.Q. Chen, T.M. Zhan, Z.Q. Jiao, Y. Sun, Z.M. Chen, S.H. Wang, Fractal dimension estimation for developing pathological brain detection system based on Minkowski-Bouligand method. IEEE Access **4**, 5937–5947 (2016). https://doi.org/10.1109/ACCESS.2016.2611530
21. Y. Gui, Hausdorff Dimension Spectrum of Self-affine Carpets Indexed by Nonlinear Fibre-coding, in *2009 International Workshop on Chaos-Fractals Theories and Applications*, (IEEE, 2009), pp. 382–386. https://doi.org/10.1109/IWCFTA.2009.86
22. E. Guariglia, Entropy and fractal antennas. Entropy **18**(3), 84 (2016). https://doi.org/10.3390/e18030084
23. C. Sevcik, A procedure to estimate the fractal dimension of waveforms. arXiv (2010) preprint arXiv:1003.5266
24. A. Petrosian, Kolmogorov complexity of finite sequences and recognition of different preictal EEG patterns, in *Proceedings Eighth IEEE Symposium on Computer-Based Medical Systems*, (IEEE, 1995), pp. 212–217. https://doi.org/10.1109/CBMS.1995.465426
25. M. Ezz-Eldin, A.A. Khalaf, H.F. Hamed, A.I. Hussein, Efficient feature-aware hybrid model of deep learning architectures for speech emotion recognition. IEEE Access **9**, 19999–20011 (2021). https://doi.org/10.1109/ACCESS.2021.3054345
26. V. Kadyan, S. Shanawazuddin, A. Singh, Developing children's speech recognition system for low resource punjabi language. Appl. Acoust. **178** (2021). https://doi.org/10.1016/j.apacoust.2021.108002

27. E. Guariglia, Spectral analysis of the Weierstrass-Mandelbrot function, in *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, (IEEE, 2017), pp. 1–6
28. C.T. Shi, Signal pattern recognition based on fractal features and machine learning. Appl. Sci. **8**(8), 1327 (2018). https://doi.org/10.3390/app8081327
29. A. Ezeiza, K.L. de Ipina, C. Hernández, N. Barroso, Enhancing the feature extraction process for automatic speech recognition with fractal dimensions. Cogn. Comput. **5**(4), 545–550 (2013). https://doi.org/10.1007/s12559-012-9165-0
30. V. Kadyan, A. Mantri, R.K. Aggarwal, A heterogeneous speech feature vectors generation approach with hybrid hmm classifiers. Int. J. Speech Technol. **20**(4), 761–769 (2017). https://doi.org/10.1007/s10772-017-9446-9
31. J. Singh, K. Kaur, Speech Enhancement for Punjabi Language Using Deep Neural Network, in *2019 International Conference on Signal Processing and Communication (ICSC)*, (IEEE, 2019), pp. 202–204. https://doi.org/10.1109/ICSC45622.2019.8938309
32. M. Qian, I. McLoughlin, W. Quo, L. Dai, Mismatched training data enhancement for automatic recognition of children's speech using DNN-HMM, in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, (IEEE, 2016), pp. 1–5. https://doi.org/10.1109/ISCSLP.2016.7918386
33. M. Manjutha, P. Subashini, M. Krishnaveni, V. Narmadha, An Optimized Cepstral Feature Selection method for Dysfluencies Classification using Tamil Speech Dataset, in *2019 IEEE International Smart Cities Conference (ISC2)*, (IEEE, 2019), pp. 671–677. https://doi.org/10.1109/ISC246665.2019.9071756
34. V. Kadyan, A. Mantri, R.K. Aggarwal, A. Singh, A comparative study of deep neural network based Punjabi-ASR system. Int. J. Speech Technol. **22**(1), 111–119 (2019). https://doi.org/10.1007/s10772-018-09577-3
35. J. Guglani, A.N. Mishra, Continuous Punjabi speech recognition model based on Kaldi ASR toolkit. Int. J. Speech Technol. **21**(2), 211–216 (2018). https://doi.org/10.1007/s10772-018-9497-6
36. K. Goyal, A. Singh, V. Kadyan, A comparison of laryngeal effect in the dialects of Punjabi language. J. Ambient. Intell. Human. Comput. (2021). https://doi.org/10.1007/s12652-021-03235-4
37. J. Guglani, A.N. Mishra, Automatic speech recognition system with pitch dependent features for Punjabi language on KALDI toolkit. Appl. Acoust. **167**, 107386 (2020). https://doi.org/10.1016/j.apacoust.2020.107386
38. G. Sreeram, K. Dhawan, K. Priyadarshi, R. Sinha, Joint Language Identification of Code-Switching Speech using Attention-based E2E Network, in *2020 International Conference on Signal Processing and Communications (SPCOM)*, (IEEE, 2020), pp. 1–5. https://doi.org/10.1109/SPCOM50965.2020.9179636
39. J. Li, L. Deng, Y. Gong, R. Haeb-Umbach, An overview of noise-robust automatic speech recognition, in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22(4), (IEEE, 2014), pp. 745–777. https://doi.org/10.1109/TASLP.2014.2304637
40. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, K. Vesely, The Kaldi speech recognition toolkit, in *IEEE 2011 workshop on automatic speech recognition and understanding (No.CONF)*, (IEEE Signal Processing Society, 2011)
41. S. Rajendran, P. Jayagopal, Preserving learnability and intelligibility at the point of care with assimilation of different speech recognition techniques. Int. J. Speech Technol. **23**, 265–276 (2020). https://doi.org/10.1007/s10772-020-09687-x

Class Diagram Generation from Text Requirements: An Application of Natural Language Processing

Abdulwahab Ali Almazroi, Laith Abualigah, Mohammed A. Alqarni, Essam H. Houssein, Ahmad Qasim Mohammad AlHamad, and Mohamed Abd Elaziz

1 Introduction

The software program development procedure is long and complex. It operates by knowing the user's needs (called requirements or specifications); this is the main point toward which the whole software program may be produced [1]. This section covers numerous arrangements and meetings until the ultimate plan of requirement terms is supplied. This file or document, described in the specification, is called the Software Requirements Specification (SRS) file [2]. The developers utilize this record (SRS) for growing and analyzing the wanted application. SRS presents

A. A. Almazroi
Department of Information Technology, University of Jeddah, College of Computing and Information Technology at Khulais, Jeddah, Saudi Arabia

L. Abualigah (✉)
Research and Innovation Department, Skyline University College, Sharjah, United Arab Emirates

Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia

M. A. Alqarni
Department of Software Engineering, University of Jeddah, College of Computer Science and Engineering, Jeddah, Saudi Arabia

E. H. Houssein
Department of Computers and Information, Minia University, Minia, Egypt

A. Q. M. AlHamad
Information System Department, Sharjah University, Sharjah, United Arab Emirates

M. A. Elaziz
Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_4

complete details about the training that must be presented, the functions, styles, and procedures that must be incorporated, and many others. This report is human-regularly occurring, but big tasks have many SRS sides and are henceforth almost infeasible for a human to observe and inspect. Consequently, the researchers of this chapter hope that they can find a new, robust strategy to address this problem [3]. Also, an optimization techniques could be used to solve this problem [4–7].

Universally, the existing techniques were employed to determine the desired diagrams from the requirements record, which is assessed into two predominant strategies: conventional-based strategies and object-orientated-primarily based procedures. Those processes are selected for getting the best out of the machine's targets and purposes, whereas the recent method is related to the item-orientated model. It describes instructions, features (attributes), and regulations (methods). It further prepares the correlation among classes if it exists [8, 9]. UML diagrams mainly combine two main components: static and dynamic design. The static design is additionally declared as a structural design that comprises class and compound structure designs (diagrams) [10, 11]. Moreover, the behavior diagrams focus on what requirements arise in the application design. Dynamic design diagrams (behavior diagrams) are designed to explain the procedures and operations of a system, which are also employed to illustrate the functionality of software services. It is a considerably time-consuming and challenging responsibility to produce designs from natural language (NL) system requirements [8].

Design patterns are usually employed to prepare the requirements document in an additional structured way; various design patterns are intercommunication design diagrams, sequence diagrams (control), and activity design diagrams [12]. Various techniques and tools have been developed to succeed in the gap among demands review (requirements) and design stage by producing object-oriented prototypes from the provided requirements [13].

In the past, data flow diagrams (DFD) have been employed to symbolize the fact stream and draw the user's demand specifications. However, within the modern age, a unified modeling language (UML) is implemented to map and display the users' needs onto statistical drift charts, which is a complete and legitimate method of supplying this information. UML is beneficial for the following types of software development methods [1]. Any software development process begins with a requirement for investigation and evaluation as described in the first section. This segment will chalk up a clarifying layout to apprehend the most challenging and troublesome tasks in programming development. Disasters made this action uncontrollable to adjust in greater backward durations of programming development. Number one, the reason for these abilities is to report the importance of using them in NLP. To overcome this, a device has been produced, which designs a semi-automatized manual for designers to create a UML class version from software specifications and use NLP strategies. This method aims to demonstrate the class diagram in a traditional and widespread configuration and further list out the connections among the instructions [2]. With all the one's representations to show the human point of view, human language is an unclear but very adaptable method; graphic designs are

the most effective; however, mathematical language is the most particular; but it needs a higher level of expertise to develop and to comprehend [14].

The records needed are usually determined and provided by an expert in computer usage. It is proved that factors of information and statistics retrieval are nearly equal precepts. Analysts can rent diverse strategies required to accumulate relevant statistics for software program evolution. This statistic defines coverage expectations in terms of layout requirements, targets, barriers, performance, and robustness standards. Consequently, a mechanism plan is needed so that it supports the automation of initiatives included in the particular life cycle steps of software engineering improvement, which also includes the class diagram that will be introduced later in this article.

These machines reflect on accomplishing purposeful and nonfunctional tasks ranging from textual statistics to enhanced visibility of utility envelopes to assist customers in assuming the management of an evolutionary process that is not dependent on the use of a particular technology. This research intends to construct a device that transforms a textual, natural language and data to a UML magnificence diagram. This tool is used to input text data that constitutes the textual input specs necessary to allow a person to understand the text. Firstly, it classifies entities' names (i.e., classes, properties, and relationships among styles of instructions). Secondly, it prepares them in a based XML document. The proposed technique is employed to achieve the magnificence diagram of the given necessity specification for a selected system. The proposed technique improves analyzability by performing an effective and fast rule to generate a class diagram from the magnificence requirements of the data set. It continues this interplay between customers with the aid of providing a familiar social-related user interface.

The remainder of the study is systematized as follows: In Sect. 2, a comprehensive review is presented for the related work to show for the reader where the research is reached in this domain. In Sect. 3, the methodology of this work is given to show all the procedure steps. In Sect. 4, experiment results and discussions are given to demonstrate the performance of the proposed method. And finally, in Sect. 5, conclusions and possible feature works are given.

2 Related Work

This section shows the relevant work and discusses its main procedures, as shown in the next subsections.

2.1 Unified Modeling Language (UML) Diagram

One of the problems that arise in the SDLC is through the requirements analysis and design. The issues faced through the first stage of the transfer process to other steps,

resulting in a high-cost process in comparison with the original method. The human language style can assist developers in determining the software specifications by changing the elements in an electronic design using UML diagrams [13].

This article is focused on producing the sequence design and activity design diagrams and utilizing the requirements by presenting them in the natural language [15]. The parser and the PSO tagger methods are used to analyze the user input, provided in the English language when selecting the procedures and expressions and others from the text.

In another paper [8], a novel technique was introduced to improve the overall processing of the NL specification, and the proposed approach detected defects in NL specifications. Moreover, the authors showed from the comparative study how well the proposed method supports even non-software engineers in editing texts for generating software engineering requirements. The obtained results in this research showed that the proposed method could speed up creating texts with more scattered defects significantly.

The proposed method in another study [12] improved the connection between traditional modeling language and conventional natural language processing. This method has been produced by Java and tested on some personal documents. Moreover, several mechanizations have been introduced to retain the models compatible and the textual designation when the rules change [8].

Another study represented the NLP mechanism, which was endeavored to help the investigation step of software improvement in an object-oriented structure [16]. This NLP procedure was created to investigate software demand texts reproduced in English and to make a combined dialogue paradigm of the prepared text that is described in a grammatical system. This system was then applied by itself with little or no direct human control to build a UML class diagram such as class design illustrating the term types specified in the given text, the relations between them, and a sequence diagram of the electric model. The specification review defined the user's needs for a specific purpose. In [15], a mechanism was proposed to obtain the figures or diagrams from the given requirement documents with powerful semantic assistance. The recommended tool converted the user modeling details into the programming specialist code; code creation was created and was available in Java. The primary aim of this works' proposed method is to illustrate the use of NLP methods for generating all kinds of UML design diagrams with code framework production in Java by performing a model tool that utilizes the NLP procedures.

Another extraction paradigm was named UML design generator from the interpretation of requirements; it was introduced to obtain the UML designs, from natural language requirements using useful NLP mechanisms. In this method, the complex wants to be processed into precise needs based on a collection of syntactic reconstitution commands. Moreover, a visualizing design of any UML design will be performed by XMI design [17].

Software specifications are an essential action of the software process; the failures at this stage will necessarily be directed to difficulties that will arise later in system configuration and implementation. The requirements are formulated in NL, with the potential for uncertainty, inconsistency, or mistake, or naturally a failure of

programmers to deal with a large volume of knowledge. This chapter proposed a new method for the NLP of requirement descriptions of the universal natural language and their automatic changing to the object-oriented interpretation system [18].

To change the human language into use-case and class diagrams, a new method was introduced to perform these operations; two states have been produced. Firstly, the recursive object design, which changes from human language to graphic style (writing). It also converts the recursive object design to the UML style [14]. Additionally, it changes the UML designs to the human language [19]. Moreover, it introduced a new system using a grammatical structure to convert the class design diagram into a common linguistic or human language. Then, the linguistic form was converted into the human language (text). The class diagram's automatic generator is introduced as an approach that aggregates the human language (NL) methods' analytical and design verification characteristics. Various models were determined to generate the class diagram. When the ideas are created, the XML metadata file was created and transmitted with a computer-aided tool to generate the UML design diagram [20, 21].

A new algorithm was introduced to facilitate the extraction rule in generating UML design diagrams from the human language produced by various people [22]. Improving the UML's regular syntax and identifying the legal issue in the initial stage will decrease the time and cost. This work also gave a symmetrical syntax model for UML diagrams and use-case diagrams [23].

Nowadays, the most powerful software applications that offer assistance to induce UML design diagrams more strongly are Rational Rose, Clever Draw, etc. Currently, these softwares are the ones recommended for the said purpose, but they have many drawbacks and disadvantages. According to the criteria and rules, the design analyst must do some paintings to deduce the general business function and follow the user necessities earlier than describing the UML designs through the usage of orthodox CASE gear. Consequently, a lot of time is wasted due to the available CASE equipment's dual nature for the necessary situation. Nowadays, every person wants a brief and reliable service. Therefore, it is transformed into a requirement that there ought to be some intelligent software applications for creating UML based totally on text to save time and resources for both the user and equipment analyst [1].

2.2 Class Diagram

Each software improvement process begins with a fundamental review. The first phase from requirement review is to generate a design that is recognized as the usual challenging and demanding applications in programming improvement. Nonperformance is presented during this action, which can be quite stubborn to change in more advanced stages of programming progress. One main objective behind such possible difficulties is to show the essential purpose of implying them

in ordinary language applications. A solution to this problem was suggested by [2], where a mechanism has been designed to provide authors semi-automatized assistance to generate a UML class design from software phrases using NLP techniques. The proposed system illustrates the class diagram in a traditional form and histories the association among the provided classes.

A method is introduced by [24] to improve the specifications of the requirements system and to obtain the class diagram from the human language needed by keeping NLP. The conditions are investigated humanly by experts to find out the class diagrams; therefore, such a computerized approach is required to overcome human errors. Requirements and class diagram generation are tools proposed to support requirements examiners and software engineering scholars explore the human language in generating the software requirements, which reached the core theories and relationships and select the class diagram. To explain the business reports as a text to generate the UML diagrams, an automatic system is introduced based on using the natural language. Users record demands in the English human language in numerous texts, and the introduced method has an uncommon ability to explain the given text record. The next step is to perform the analysis stage and extraction stage; the system described various UML design diagrams, class design diagrams, and sequence design diagrams. The proposed work gave a stable and active method to generate UML diagrams [1].

Researchers proposed to change the software specifications to the object-oriented model [25, 26]. Software requirements specifications are described in the human language by various complex situations such as style converted to UML class diagrams by particular change conditions. The change process is utilized using the syntax interpretation for the software specifications. The authors aimed to use the NL process to change the software elements to the formal term. A new method is utilized to examine the command and select the class design diagram from human terms to manage the NLP strategy [27].

Generation of human language specification from UML class design diagrams is introduced to solve the problem of the connotation between the human language and computer language [28]. In the beginning, the class of data language is employed to analyze the class design based on examining many cases to produce rules for reducing the uncertainties in the part of human language utilized in UML. To obtain precise UML's lexical systems and generate grammatical decisions, the WorldNet ontology is employed. To interpret the software specification and create a merged dialogue model specified in a grammatical arrangement [15]. The selected material that worked to produce the UML model includes the class diagram describing the terms recorded in the human language regarding the classes. The demand can also change the user presentation into segments of the Java style source code [3].

2.3 ER Diagrams

In the entity-relationship (ER) information design (which is introduced to assist in generating the database), the structure of the ER designs makes it a complex job to

determine the ER for both students and designers. To tackle this issue and generate the ER diagram components from the human language terms, various solutions are suggested by using the NLP. An approach is introduced using human language records to make the ER design. To generate the terms of requirement specifications, the structural approach is applied based on specific rules [29].

Requirements interpreters recognized a conceptual design to be a vital artifact generated through the requirements review stage of an SDLC (software development life cycle) [30]. A conceptual design is a visible model of the term's requirements specialty in the center. Because of its visible nature, the design works as a stand for the discussion of demands by stakeholders. It allows requirements investigators to improve the additional functional specifications. Conceptual rules usually result in class design diagrams through the configuration and achievement stages of the software plan. Additionally, a somewhat mechanical conceptual design can keep enough time through the analysis stage by accelerating the rule of the graphical interface, conception, and visualization.

Another work introduced a new scheme to produce a mental concept design from useful terms, recorded in NL in a computerized model [18]. Classes' diagrams and their connections are automatically recognized by the working requirements. This description depends on the interpretation of the syntactic generates of judgments and object-oriented systems of purpose. EER (extended entity-relationship) systems are combined toward the class associations. Optimization procedures are employed to recognize objects through a post-processing step, and the last conceptual illustration is provided. The advantage of written dominions—mixed with several rules—is used to determine a class relationship that gives a desirable method to the generation of the object terms in the design. The work demonstrates the model making process of running a current case study. It terminates with an assessment of the suitability of the proposed method for the specification review and investigation. The interpretation is compared with two standard issued models in the literature and conceptual forms designed by individuals for different assessment parameters.

The selected words outlined in ER design details such as entities, relationships, and attributes. In this work, a new method to generate a practical design from functional requirements is proposed, presented in the human language using a computerized system. Classes and relationships are recognized from the functional requirements—these classification rules depend on the creation of the sentence language and the principle of the object-oriented model. Extended ER systems are mixed with the class associations. The optimizations are executed to the classified entities, while a post-processing level and the last conceptual arrangement are provided. The utilization of recorded mandates, mixed with commands to define class relations, provides a desirable approach to generate the terms in the standard form [18]. The authors of this research keep in their mind the magnificence class diagram for numerous motives, which can be as follows:

- A few pieces of research have been used and employed to convert human language requirements into class design diagrams for development purposes.
- Class diagrams are the main part of the design and analysis of any system, and the rest of the methods are taken from the class diagrams. Moreover, they comprise

most of the information, although not in characteristics and specifications needed in systems' requirement features.
- Use-case design diagrams are ignored at this level because they provide details that are mainly formulated in the human language. Subsequently, it does require much production and is fully suitable to be understood by users.
- The knowledge involved in translation use-case diagrams is similar to the procedures associated with the class diagrams. They are mostly used to discover where terms collaborate and interact and how they convert from one circumstance to another.

Reducing the time and work in object-oriented modeling is vital in the software engineering domain and NLP. Therefore, to solve the problem of producing class diagrams and provide a robust solution, a proper framework is required to assist the users and software engineers [31–33]. The conducted work was domain specific; however, it could be developed quickly in years to come based on the given requirements. The planned application currently covers the ability to map user requirements once one reads the provided requirements in plain text and generates a set of UML design diagrams as an activity, component diagram, sequence diagram, use-case diagram, and class diagram. An incorporated improvement enhancement would further give sufficient input, user interaction, and output. Other natural language processing techniques can be used in the future [34–39].

3 The Proposed Method

The proposed method is discussed in this section to solve the system requirement translation; the proposed approach converts the users' demands (requirements) into the UML class design diagram. The proposed method, as shown in Fig. 1, is proposed to address these processes carefully for the users. The requirement terms are elements created by the user, which describes that the system employs pure human presentation.

3.1 Main Procedures of the Proposed Method

In the proposed method, the complexity is decreased by disseminating applications in more inadequate modules as follows.

3.1.1 Tokenization and Sentence Selection

In the first step, tokenization will determine different stop words in the sentence, such as a, the, in, you, me, and no. Various approaches have been proposed to identify the stop words. However, these approaches are not useful in obtaining

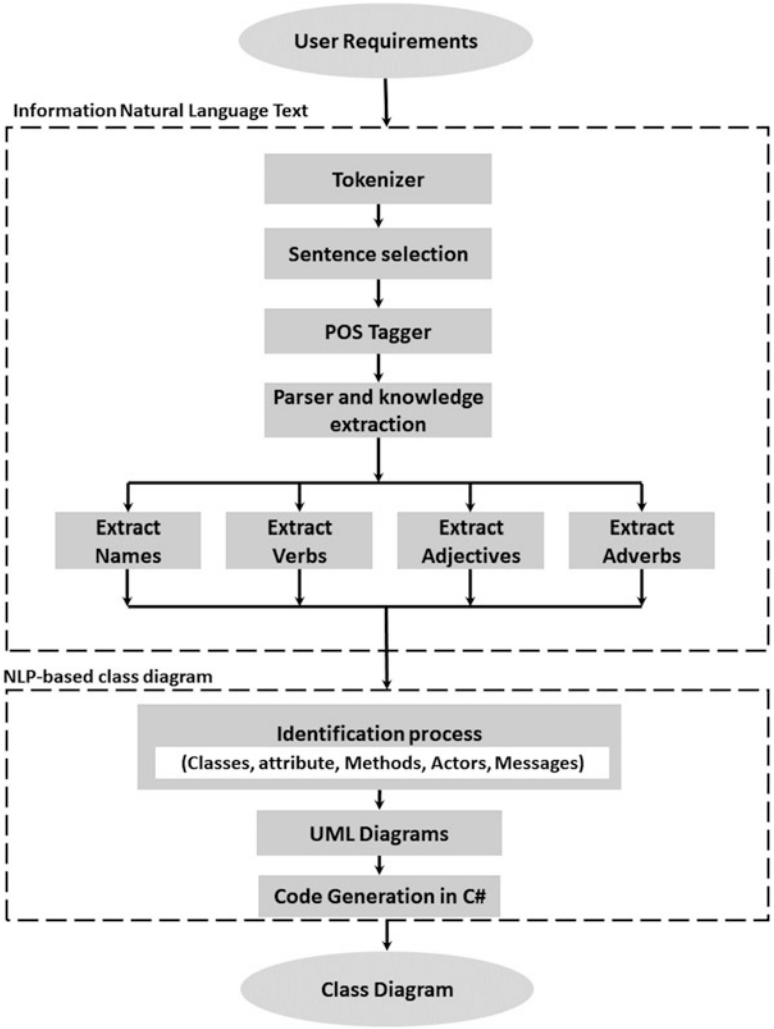

Fig. 1 The proposed method

knowledge. In this research, authors will collect the most common stop words, and then a matching system will extract these words from the text [15]. Moreover, the authors of this research will use the stemming algorithm to get the root for each word for more examination by removing the affixes and suffixes.

3.1.2 Part-of-Speech Tagger and Knowledge Extraction

Here, the method will parse the output to find out the nouns, verbs, adjectives, and adverbs. This research suggests using the Open NLP POS Tagger to do this analysis.

3.1.3 Class Diagram Using NLP

This section presents a new method to automatically create the required class diagrams. The input of this process will be a regular text and structural text, while the output will provide the class design diagram with its class and attributes and relationship. The interclass connections can arrange toward association, aggregation, and generalization. The association rules can organize into three fundamental relations: one to one, one to many, and many to many [40].

The introduced technique implements the NPL to explore instructions of the case; that is, we distinguish prime classes of the location as a beginning issue, for which we typically are very favorable, and decide kinds of classes which can be linked with the known ones. The strength of this procedure is that it defines characters and associates in one motion. Ere going into this step, we perform the following fundamental additives of the approach and how they are presented in this research to help in defining its justification.

Class Identification

Both the POS tagger and the sentence analysis provide fundamental solutions. On the other hand, the semantic network approach and word sense are explicitly implemented to get the original candidates.

Relationship Identification

To extract the relationship, it utilizes a linkage range to define all notion couples with robust grammatical relations within the sentence. We provide several weights for all theory set to show how robust the connection is based on the components the notions work as in the determination.

Attribute Identification

The attribute identification system describes the class attributes followed by the two notions, which are obtained to be completely correlated together; we need to decide if the theories are related to the class attribute or the class.

Naming Relationship

Naming relationship implements the form semantic; this helps to obtain an association. This model uses several relationship recognition approaches to classify the relationship type like one to one, one to many, and many to many.

The proposed approach uses the repetition procedure to determine the classes and the characteristics. It picks one concept from the particular set with the highest relationship rate and is associated with the specified classes. The relevance rate is an indicator of how the new theory semantically joins with all other purposes of the set of nominee plans. If the correspondence score is smaller than the association's inception, the course ends. Differently, it will include the idea of the collection of types or the set of characteristics to the number of characteristics the concept has. Lastly, to utilize the previous outcome of the analysis, UML design class diagrams are produced, and the code template will be generated by C#. The approach manipulates the graphical depiction of the UML designs and allows the consumer to rename classes, add, delete, and connections within the produced layout. As a member of UI, the concept supervision UI is a number one interface that allows the person to view, combine, exchange, and prepare concepts and connections. Customers can upload new designs and exchange the concept type. The concept management approach facilitates the consumer with the ability to make the processing because the person needs.

4 Experiments and Results

4.1 Case Study

This section presents with results the application of the conceptual model production procedure. The given requirements specification and details are obtained from the ATM dilemma statement [41]. The first part is manually changed to eliminate kinds of pronouns, references, and wh pronouns (who, whose, whom, whatever, etc.). It is further adjusted to assure regular utilization of a term for a regular function, i.e., one function per conversation (Fig. 2).

The system must support a computerized banking network that includes both human cashiers and ATMs. The computerized banking network will be shared by a consortium of banks. Each bank provides a computer that maintains the bank's accounts and processes transactions against the accounts. Cashier stations are owned by individual banks and communicate directly with the bank's computers. Human cashiers enter the account data and transaction data. An ATM communicates with a central computer. The central computer clears transactions with the banks. An ATM accepts a cash card and interacts with the user. An ATM communicates with the central computer to carry out transactions. An ATM dispenses cash, and prints receipts. The system requires appropriate record-keeping and security provisions. The system must handle concurrent access to the same account correctly. The banks will provide the bank's own software for the bank's own computers.

Fig. 2 Modified ATM statement from Rumbaugh's ATM problem [41]

4.2 Conceptual Modeling

In the preprocessing stage, the components are divided into decisions. The outcome of the syntactic characteristic extraction provides the collection of verbs, adjectives, proper, adverbs, and collective nouns. The ultimate collection that classifies the terms into parts of speech (POS) in the journey is shown in Fig. 3: the ultimate collection that classifies the words into POS.

Within the last listing of additives, ATM has taken into consideration a collective noun class. A noun that transpires as a novel noun command lets in fashionable noun statistics if employed in that manner additionally one time. In this instance, the

Adjectives	Adverbs	Nouns	Proper Nouns	Common Nouns
'computerized', ,'central', 'human', 'own', 'appropriate', 'same', 'concurrent', 'individual'	'directly', 'correctly'	'receipt', 'station', 'cashier', 'bank', 'computer', 'user', 'access' , 'data', 'card', 'transaction', 'network', 'security', 'record-keeping', 'consortium' , 'provision', 'atm', 'system', 'account', 'banking', 'cash', 'software'	None	'receipt', 'bank' , 'cashier', 'user', 'consortium', 'security', 'card', 'account', 'transaction', 'computer', 'atm' , 'network', 'system', 'access', 'banking', 'provision' , 'station', 'cash', 'software', 'data', 'record-keeping'

Fig. 3 The ultimate collection that classifies the words into POS

utilization of "ATM" in the beginning sentence "The machines have to aid a computerized banking network that includes each human cashiers and ATMs" has made ATM recollect a preferred noun creation.

The design components are decided on and selected. The written dominions of every sentence are moved throughout each application component (rule). The phase of the ensuing software component defines what attributes, lessons, and family members are taken for an expression. As an example, for the dedication, an ATM interacts with a primary computer; the order applied to look at for commands for this expression is presented in Fig. 4.

Every statement is prepared by various rules based on the statement construction, the appearance of connections, prepositional problems, gerunds, etc. The obtained outcomes of the design components taking out for the provided statement, next all the given rules are performed, are:

4.2.1 Classes

This list contains a set of terms (atm, system, central_computer, consortium, bank, computerized_banking_network, individual_bank, cashier_station, cashier, computer).

4.2.2 Class Relations

The outline of relationships and relationship fame produced are presented in Fig. 5. We utilized the rule for selecting the scheme components as introduced in the end column. The benefits of the application of mandate parsing are obvious from the evidence that the statement is divided into parts, and the application rules are according to the fragmentation of the state structures, instead of the whole statement. The generic connections are then grouped based on their character, and small classes and connections are treated to omit them by introducing as operations and attributes, respectively.

4.2.3 Aggregation Classification

Computerized_banking_network>cashier (Aggregation)

Computerized_banking_network>atm (Aggregation)

The abovementioned are a couple of relationships, from Table 3 convert aggregations relationship.

4.2.4 Composition Classification

Consortium>bank (Composition)

Typed dependencies	Rules checked	Rule results and further actions	Trace of low level actions in rule results processing.
det ATM An	Ignored		
nsubj communicates ATM	Check 'Basic Rule'	obj does not exist for same verb Fails	
	Check 'Intransitive Verb Rule'	Fails, due to presence of prepositional object	
det computer a	Ignored.		
amod computer central	Ignored.		
prep_with communicates computer	Check 'Prepositional subject and object'	Pass	
		class(ATM)	Stem (atm) = atm
			No compound nouns
			No adjective modifiers
		Create atm class	
		class(computer)	
			Stem (computer) = computer
			No compound nouns
			Adjective Modifier found
		class(central_computer)	No classification for 'central'
			Append to class name
			computer->central_computer
		Not created, as this is an object	

Fig. 4 A case sample using trace plan of implementation rules

4.2.5 Generalization Classification

We have no cases of generalization inside the provided record. An assertion, "A pc is a digital tool," could turn out to be a standard generalization of the processor type.

No.	Sentence No.	Class 1	Class 2 (may not exist)	Name of relation	Rule used
1.	0	computerized_banking_network	atm	Includes	• Compound noun generator
					• Elementary subject object rule
2.	0	system	computerized_banking_network	support	• Compound noun generator
					• Elementary subject object rule
3.	0	computerized_banking_network	Cashier	Includes	• Compound noun generator
					• Elementary subject object rule
					• Adjective classifier–human denotes quality
					• cashier added as class, with attribute quality = human
4.	1	computerized_banking_network	consortium	shared_by	• Compound noun generator
					• Passive subject with agent
5.	1	consortium	bank	of	• Secondary 'of' preposition following agent in passive subject (composition)
6.	2	bank	computer	provides	• Elementary subject object rule

Fig. 5 Sample application with connections after design component extraction

Table 3 Performance evaluation

Case study				
Methods		ATM [41]	Course registration [45]	EFP [46]
Without implicit information or assumptions				
Overall	Recall (%)	83.33	100	93
	Precision (%)	100	80	88
	Over_SR (%)	100	100	93
Association	Recall (%)	80	100	86
	Precision (%)	100	100	75
	Over_SR (%)	120	0	171
Composition	Recall (%)	100	100	100
	Precision (%)	100	75	100
	Over_SR (%)	0	133	100
With implicit information/assumptions				
Overall	Recall (%)	50	57	61
	Precision (%)	100	80	88
	Over_SR (%)	60	57	61
Generalization	Recall (%)	44	100	100
	Precision (%)	100	100	100
	Over_SR (%)	67	0	100
Association	Recall (%)	100	50	50
	Precision (%)	100	75	75
	Over_SR (%)	0	67	100
Proposed method				
Overall	Recall (%)	85	100	95
	Precision (%)	100	80	100
	Over_SR (%)	100	100	94
Generalization	Recall (%)	49	100	100
	Precision (%)	100	100	100
	Over_SR (%)	69	0	100
Association	Recall (%)	100	60	55
	Precision (%)	100	78	80
	Over_SR (%)	0	68	100

4.2.6 Trivial Associations to Attributes

The hint for the cause of small classes is as right here, together with the objects:

Evaluating relationship: bank>computer (has)

Association: non-trivial (has\have) association retained bank; computer

In the connection bank owns a computer, the bank is preserved as a distinct state (class) due to the additional notice of the bank class.

Evaluating relationship: bank>account (has)

Attribute (has\have) association moved to attribute bank: account

Evaluating relationship: bank>software (has)

Attribute (has\have) association moved to attribute bank: software

Not *account* or *software* is additionally defined and does not survive as classes; therefore, it is designed as characteristics of the class *bank*.

4.2.7 Trivial Relations to Operations

Associations that hook up with nonexisting properties convert to magnificence moves due to the training that they lead to small references. In the following hint, the second item does now not exist as a class.

Evaluating relation →atm: cash (dispenses)
Travel association→class opern: atm: dispenses_cash
Evaluating relation → systemr: account (handl_to)
Travel association→class opern: system: handle_to_account
Evaluating relation →computer: account (processes_against)
Travel association→class opern: computer: processes_against_account
Evaluating relation →atm: cash_card (accepts)
Travel association→class opern: atm: accepts_cash_card
Evaluating relation →atm: cash (prints)
Travel association→class opern: atm: prints_cash
Evaluating relation →system: conqurent_access (handle)
Travel association→class opern: system: conqurent_access_handle
Evaluating relation → central_computer: transaction (clears)
Travel association→class opern: central_computer: clears_transaction
Evaluating relation → system: security_provision (requires)
Travel association→class opern: system: requires_security_provision

4.2.8 Association Classification

The rest of the given relationships after working through all the earlier actions are maintained as associations.

Evaluating relation → bank: bank (provided_software)
Association→ bank: provide_software_bank
Evaluating relation → computerized_banking_network: consortium (shared_by)
Association→ computerized_banking_network: shared_by_ consortium
Evaluating relation → computer: bank (maintains_accounts)
Association→ computer: maintains_accounts_bank

4.2.9 Trivial Class Removal

At this level, a pre-known vocabulary, and words to be extracted are worried. Those phrases are used far from the menu of classes and relationships. Inside the ATM version, the word “gadget” can be expressed as a small class to be eliminated. The

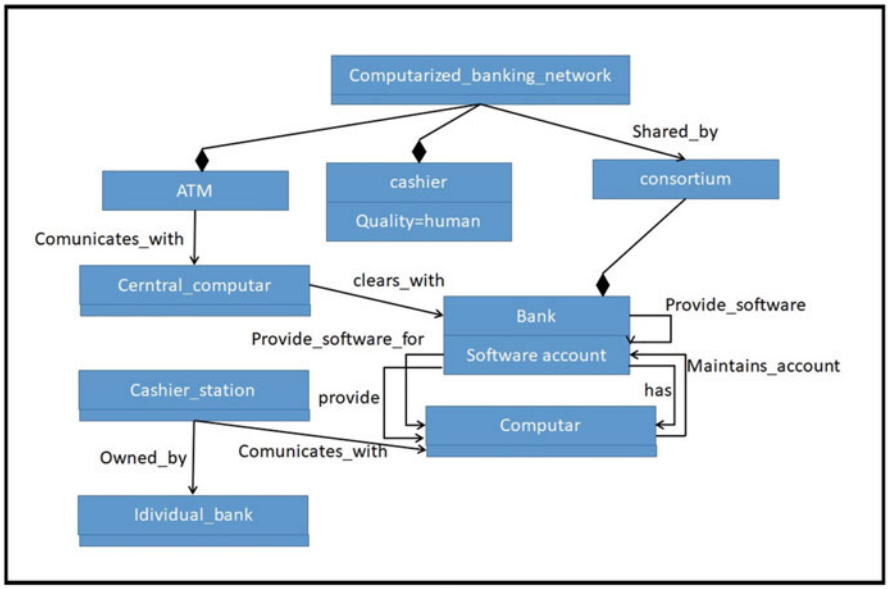

Fig. 6 Conceptual model with only attributes, relations, and classes

received results later, the fulfillment of all measures listed in Table 4. In Table 4, "a" refers to trivial relations moved to operations, "b" refers to trivial associations moved to attributes, "c" refers to aggregation, "d" refers to a composition, and "e" refers to the remaining association.

The conceptual model is performed using theories from Bhala in [42] to produce practical capability. The theoretical principle is built by programmatically generating code in the DOT criticism, which is a portion of visible visualization software. The normalized frequency of occurrence defines the priority attached to each class. The standardized rate is done by the rate of reference concerning class name in the method of the production of the object elements. The obtained results are shown in Fig. 6. In the illustrated figure, ATM and bank are the top couple classes that have been emerged (Table 1).

4.3 Performance Evaluation

The effectiveness assessment of this method changed into trouble due to the fact that we have no explanation of a "right" conceptual version. The theoretical models that are commonly accomplished have been determined to consist of extensively brought enjoyment via the investigator. Unique know-how inside the conceptual version is much like lessons and/or members of the family that are not unique inside the textual content of the call for.

Table 1 Classification results of the relationships

S no.	Class 1	Class 2	Relation name
1	Computerized_banking_network (c)	Atm (c)	Includes (c)
2	Computerized_banking_network (c)	Cashier (c)	Includes (c)
3	Computerized_banking_network (e)	Consortium (e)	Shared_by (e)
4	Consortium (d)	Bank (d)	Of (d)
5	Bank (e)	Computer (e)	Provides (e)
6	Bank (b)	Account (b)	Has (b)
7	Computer (e)	Bank (e)	Maintains_accounts (e)
8	Computer (e)	Bank (e)	Processes_transaction (e)
9	Computer (a)	Account (a)	Processes_against (a)
10	Cashier_station (e)	Individual_bank (e)	Owened_by (e)
11	Cashier_station (e)	Computer (e)	Communicates_with (e)
12	Cashier	Account_data (a) (a)	Enter (a)
13	Cashier (a)	Transaction_data (a)	Enter (a)
14	Atm (e)	Central_computer (e)	Communicates_with (e)
15	Central_computer (a)	Transaction (a)	Clears (a)
16	Central_computer (e)	Bank (e)	Clears_with (e)
17	Atm (a)	Cash_card (a)	Accepts (a)
18	Atm (a)	User (a)	Interacts_with (a)
19	Atm (e)	Central_computer (e)	Communicates_with (e)
20	Atm (a)	Receipt (a)	Prints (a)
21	Atm (a)	Cash (a)	Dispenses (a)
22	Bank (e)	Bank (e)	Provide_software (e)
23	Bank (e)	Computer (e)	Has (e)
24	Bank (b)	Software (b)	Has (b)
25	Bank (e)	Computer (e)	Provide_software_for (e)

For example, in the ATM case, *Bank owns Bank Computer* describes a particular opinion on a connection. At the same time, the class *Remote Transaction* is a constitutional theory that a specific class endures, though it is not declared in the requirements text.

4.4 Evaluation Criteria

In this part, evaluation criteria are utilized to analyze the proposed automatically created conceptual model with a conventional available design or a human design model in the literature [38, 43, 44]. The used criteria are:

1. Recall percentage means the capacity of the computerization to create all classes, as shown in Eq. (1).

$$Recall = \frac{Ncorrect}{Ncorrect + Nmissing} \tag{1}$$

In Eq. (1), *Ncorrect* means the number of true classes recognized; *Nmissing* means the number of classes selected by the human expert and ignored by the proposed conceptual method.

2. Precision means the accuracy, or the relevance of the categories recognized in the proposed conceptual model, as shown in Eq. (2).

$$Precision = \frac{Ncorrect}{Ncorrect + Nincorrect} \tag{2}$$

In Eq. (2), *Nincorrect* means the number of correct levels classified as wrong.

3. The over-specification rate (Over_SR) means the number of useless but right classes that the computerization process adds in the created conceptual model by the proposed, as shown in Eq. (3).

$$Over\_SR = \frac{extra\ (valid)}{Ncorrect + Nmissing} \tag{3}$$

In Eq. (3), *extra*(*valid*) is the amount of correct additional classes regained.

No matter the truth that the version is not intended for assuming unique expertise, we investigate it to shape it with human-created regulations. To determine this, we advocate any other variable. Implicit means the number of classes that are calculated, which can be proper and correct, but no longer reported within the necessities text. The equations applied to assign certain information are as follows:

$$Recall\_implicit = \frac{Ncorrect}{Ncorrect + Nmissing + Nimplicit} \tag{4}$$

$$Precision\_implicit = \frac{Ncorrect}{Ncorrect + Nincorrect} \tag{5}$$

$$Over\_SR\_implicit = \frac{Nextra\ (valid)}{Ncorrect + Nmissing + Nimplicit} \tag{6}$$

While exceptional reference rates are possible for evaluating those who do not constitute *Nimplicit*, published objective values do not survive for any of the

criteria. Therefore, for the goal of the testing process, we set advantageous positions to assess the achievement. Precision and recall can be as powerful as feasible (high) to correctly express the objective model. *Over_SR* should be low to evade attaching excessively various extra features.

4.5 *Results of Conceptual Class Modeling Versus Other Comparative Standard Models*

Table 2 presents the obtained results of the achievement measures versus various case studies. These cases were employed by several researchers to illustrate the production of class diagrams, and similar results are obtained in the corresponding references. It is observed that these conventional designs were not developed for the requirement investigation, nor generated automatically, and therefore include a component of human knowledge, which our implementation requires.

The drop-in precision illustrates the significance of human judgment to add related instructions. Our approach blended a few lessons that can be mentioned wrong. The recall measure could be very high due to all used instructions are often diagnosed, candidate lessons. Nonetheless, the over-specification weight has to continue to be at a low price, and it offers excessive values, even though elevated over-specification charges purpose visual muddle in the produced models.

The proposed method got better results in all studied cases according to the used evaluation measure, which means that the proposed method can generate the classes accurately than other comparative methods. The recall values were similar for all comparative methods. However, the precision values were better obtained by the

Table 2 Evaluation results

Case study		ATM [41]	Course registration [45]	EFP [46]
Methods				
Without implicit information or assumptions	Recall (%)	100	100	100
	Precision (%)	91.67	81.82	94.44
	Over_SR (%)	9.09	22.22	41.18
With implicit information/ assumptions	Recall (%)	91.67	100	85
	Precision (%)	91.67	81.82	94.44
	Over_SR (%)	8.33	22.22	35
Proposed method	Recall (%)	100	100	100
	Precision (%)	92.54	85.32	95
	Over_SR (%)	8.02	20.15	33

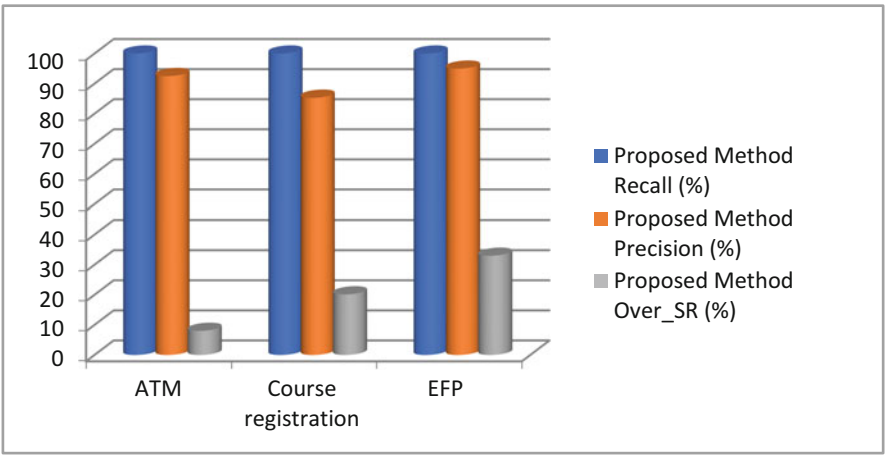

Fig. 7 Statistical analysis results

proposed method. In addition, the Over_SR values were better achieved by the proposed method compared to other methods. We concluded from the mentioned results that the proposed method got better results overall.

Because of the need for a standard text and specialty models, the experiment was carried on individual subjects. The obtained results of the final CASE tools lab test were used for reference. Figure 7 presents the final statistical results.

4.6 Performance Evaluation

Precision, recall, and over-specification are utilized to assess the effectiveness of the proposed method for the connections between the classes, as presented in Table 3. The obtained results versus human subjects additionally give comparable outcomes for relations. The relationships that are associations show that the over-SR is very high when connected to standard results. Generally, the proposed method got better results. The results are comparable for relations. The connections that are associations confirm that the over-SR is very powerful when compared to standard outcomes.

5 Conclusion and Future Work

This chapter presents a new technique to improve the procedure of creating the UML diagrams by employing natural language processing, which will assist the software developers in analyzing the software requirements with fewer errors and in an

effective way. The proposed approach utilizes the parser analysis and speech (POS) tagger to investigate and analyze the user requirements listed by the user in the English language. The acquired results revealed that the proposed method received good results than other methods published in the literature. The proposed approach gave better analysis for the given requirements and better diagram presentation, helping the software engineers. As a result, the recall measure is highly popular because all available classes are regularly recognized as candidate classes and are not subject to the recall process. Nevertheless, the over-specification weight should remain at a low rate, and it gives high values, even though high over-specification costs result in visual clutter in the created models. The proposed method can generate better results in all studied cases by using evaluation measures, which means that the proposed method can generate the classes more accurately than other comparative methods. The recall values were similar for all the relevant techniques. However, the precision values were better obtained by the proposed method. In addition, the Over_SR values were better achieved by the proposed method compared to other methods. We concluded from the mentioned results that the proposed method got better results overall. A different aspect worth further investigation is the dynamic nature of the software requirements. This can be accomplished by changing the language code into textual observations using Translatable Unified Modeling Language and Model-Driven Architecture. This will later be mixed with NL detail of the difficulty machines as they perform a vital function in the object's rules.

References

1. I.S. Bajwa, M.A. Choudhary, Natural language processing based automated system for uml diagrams generation, in *The 18th Saudi National Computer Conf. on computer science (NCC18)*, (The Saudi Computer Society (SCS), Riyadh, Saudi Arabia, 2006)
2. S.K. Kar, *Generation of UML class diagram from software requirement specification using natural language processing* (NIT, Rourkela, 2014)
3. S. Amdouni, W.B.A. Karaa, S. Bouabid, Semantic annotation of requirements for automatic UML class diagram generation. Int. J. Comput. Sci. Iss. **8**(3) (2011)
4. L. Abualigah et al., *The arithmetic optimization algorithm*. Comput. Method. Appl. Mech. Eng. **376**, 113609 (2021)
5. L. Abualigah, A. Diabat, *Advances in sine cosine algorithm: A comprehensive survey*. Artif. Intel. Rev. **54**, 2567–2608 (2021)
6. L. Abualigah, Group search optimizer: A nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. Neural Comput. Appl. **33**, 2949–2972 (2021)
7. L. Abualigah, A. Diabat, A comprehensive survey of the Grasshopper optimization algorithm: Results, variants, and applications. Neural Comput. Appl. **32**, 15533–15556 (2020)
8. M. Landhäußer, S.J. Körner, W.F. Tichy, From requirements to UML models and back: How automatic processing of text can support requirements engineering. Softw. Qual. J. **22**(1), 121–149 (2014)
9. Y. Jaafar, K. Bouzoubaa, *A Survey and Comparative Study of Arabic NLP Architectures*, in *Intelligent Natural Language Processing: Trends and Applications*, (Springer, 2018), pp. 585–610

10. R. Platt, N. Thompson, *The Past, Present, and Future of UML*, in *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics*, (IGI Global, 2019), pp. 1452–1460
11. Z.A. Hamza, M. Hammad, Generating UML Use Case Models from Software Requirements Using Natural Language Processing, in *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, (IEEE, 2019)
12. F. Friedrich, J. Mendling, F. Puhlmann, Process Model Generation from Natural Language Text, in *International Conference on Advanced Information Systems Engineering*, (Springer, 2011)
13. S. Gulia, T. Choudhury, An Efficient Automated Design to Generate UML Diagram from Natural Language Specifications, in *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, (IEEE, 2016)
14. L. Chen, Y. Zeng, *Automatic Generation of UML Diagrams from Product Requirements Described by Natural Language*, in *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, (ASME, 2009)
15. S.K. Shinde, V. Bhojane, P.Y. Mahajan, *NLP based object oriented analysis and design from requirement specification*. *Int. J. Comput. Appl.* ***47***(*21*), *30–34* (2012)
16. S. Bhagat et al., Class diagram extraction using NLP. Semantic Scholar **2**, 125 (2012)
17. D.K. Deeptimahanti, M.A. Babar, An Automated Tool for Generating UML Models from Natural Language Requirements, in *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering*, (IEEE Computer Society, 2009)
18. V.B.R.V. Sagar, S. Abirami, Conceptual modeling of natural language functional requirements. J. Syst. Softw. **88**, 25–41 (2014)
19. H. Burden, R. Heldal, Natural Language Generation from Class Diagrams, in *Proceedings of the 8th International Workshop on Model-Driven Engineering, Verification and Validation*, (ACM, 2011)
20. W. Ben Abdessalem Karaa et al., Automatic Builder of Class Diagram (ABCD): An Application of UML Generation from Functional Requirements. Softw. Pract. Exp. **46**(11), 1443–1458 (2016)
21. P. More, R. Phalnikar, Foundation of Computer Science, Generating UML diagrams from natural language specifications. Semantic Scholar **1**(8), 19–23 (2012)
22. A. Tazin, UML Class Diagram Composition Using Software Requirements Specifications, in *MODELS (Satellite Events)*, (Semantic Scholar, 2017)
23. J. Chanda et al., *Traceability of Requirements and Consistency Verification of UML Use Case, Activity and Class Diagram: A Formal Approach*, in *2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS)*, (IEEE, 2009)
24. M. Ibrahim, R. Ahmad, Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques, in *2010 Second International Conference on Computer Research and Development*, (IEEE, 2010)
25. V. Adhav et al., *Class Diagram Extraction from Textual Requirements Using NLP Techniques* (IEEE, 2010)
26. Joshi, S. Deshpande D, Textual Requirement Analysis for UML Diagram Extraction by Using NLP. Semantic Scholar 2012. 50(8): p. 42–46
27. H. Herchi, W.B. Abdessalem, *From User Requirements to UML Class Diagram* (Semantic Scholar, 2012)
28. F. Meziane, N. Athanasakis, S. Ananiadou, *Generating natural language specifications from UML class diagrams*. *Req. Eng.* **13**(1), 1–18 (2008)
29. E.S. Btoush, M. Hammad, *Generating ER diagrams from requirement specifications based on natural language processing*. *Int. J. Database Theory Appl.* **8**(2), 61–70 (2015)
30. P. Achimugu et al., A systematic literature review of software requirements prioritization research. Inf. Softw. Technol. **56**(6), 568–585 (2014)
31. J. Martin, J.J. Odell, *Object-oriented Methods* (Prentice Hall PTR, 1994)

32. A. Nanthaamornphong, A. Leatongkam, Extended ForUML for Automatic Generation of UML Sequence Diagrams from Object-Oriented Fortran. Sci. Program. **2019** (2019)
33. I. Drave et al., Semantic Differencing of Statecharts for Object-Oriented Systems, in *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development*, (SCITEPRESS-Science and Technology Publications, Setúbal, 2019)
34. L. Abualigah et al., Advances in Meta-Heuristic Optimization Algorithms in Big Data Text Clustering. Electronics **10**(2), 101 (2021)
35. L. Abualigah et al., Nature-inspired optimization algorithms for text document clustering—a comprehensive analysis. Algorithms **13**(12), 345 (2020)
36. L. Abualigah et al., A parallel hybrid krill herd algorithm for feature selection. Int. J. Mach. Learn. Cybern., 1–24 (2020)
37. L.M. Abualigah et al., An improved b-hill climbing optimization technique for solving the text documents clustering problem. Curr. Med. Imag. **16**(4), 296–306 (2020)
38. L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J. Supercomput. **73**(11), 4773–4795 (2017)
39. L.M. Abualigah et al., Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Expert Syst. Appl. **84**, 24–36 (2017)
40. X. Zhou, H. Han, Survey of Word Sense Disambiguation Approaches, in *FLAIRS Conference*, (Marshall University, 2005)
41. J. Rumbaugh et al., *Object-oriented Modeling and Design*, vol 199 (Prentice-Hall Englewood Cliffs, NJ, 1991)
42. R.V. Bhala, T. Mala, S. Abirami, Effective Visualization of Conceptual Class Diagrams, in *2012 International Conference on Recent Advances in Computing and Software Systems*, (IEEE, 2012)
43. L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm. J. Comput. Sci. **25**, 456–466 (2018)
44. L.M.Q. Abualigah, *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering* (Springer, 2019)
45. W. Kurt, O.M.T. Applying, *A Practical Step-by-Step Guide to Using the Object Modeling Technique* (SIGS Books, New York, 1995)
46. P. Swithinbank et al., *Patterns: Model-Driven Development Using IBM Rational Software Architect* (IBM Corp, 2004)

Semantic Similarity and Paraphrase Identification for Malayalam Using Deep Autoencoders

R. Praveena, M. Anand Kumar, and K. P. Soman

1 Introduction

The paraphrase identification system has to identify whether two sentences express similar meaning or not. Paraphrase identification finds many essential applications in Natural Language Processing (NLP), like machine translation evaluation, text summarization, plagiarism detection, information retrieval, question answering, etc. Deep semantic understanding is necessary for getting better performances for the problems like paraphrase identification. The proposed work explained in this chapter shows paraphrase identification for Malayalam on two levels. In the first level, the system identifies only paraphrases and non-paraphrases, but in the second level, the system is upgraded to identify semi-equivalent paraphrases and the other two. The formal definition for paraphrases, non-paraphrases, and semi-equivalent paraphrases is given below. An alternate representation of the existing sentence produces paraphrases. Thus, although they are in a different form, any two sentences having the same meaning are known as paraphrases. Consider the example given below; the sentences S_1 and S_2 are paraphrases, which justifies the semantic overlapping of sentences S_1 and S_2 is high.

R. Praveena · K. P. Soman
Center for Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore, India

M. Anand Kumar (✉)
Department of Information Technology, National Institute of Technology Karnataka (NITK), Surathkal, Karnataka, India
e-mail: m_anandkumar@nitk.edu.in

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_5

S_1:

അജ്മാനിലെബഹുനിലപാര്‍പ്പിടസമുച്ചയത്തിലന്തീപ്പിടിത്തം.

[Huge fire erupted at a residential building in Ajman]

S_2:

അജ്മാനിലെബഹുനിലക്കെട്ടിടത്തില് വന് അഗ്നിബാധ.

[Fire breaks out at a multistoried building in Ajman]

A pair of sentences that do not convey the completely same meaning is termed non-paraphrases. Sentences S_3 and S_4 are not paraphrases; even if the sentences contain the same words, it does not imply the same context.

S_3:

നാല്പത്തിനാലാമിനിറ്റിൽ ജർമ്മനിഅർഹിച്ചലീഡ്സ്വന്തമാക്കി.

[Germany gained their deserving lead in 44th minute]

S_4:

സ്വന്തംതട്ടകത്തിലാകിക്കുതുടക്കമാണ്ജർമനിക്കുകിട്ടിയത്.

[Germany got an excellent kick off at their home town]

Given a pair of sentence S_5 and S_6, where one among them carries more information than the other along with the contents of the later one, those two sentences belong to semi-equivalent. Sentences S5 and S6 are semi-equivalent paraphrases.

S_5:

ദാദ്രിസംഭവത്തിലോറസ്കിപ്പോര്‍ട്ടിനെചോദ്യംചെയ്തഅഖിലേഷ്യാദവ്.

[Akhilesh Yadav expressed suspicions about Forensics report on Dadri Incident]

S_6:

ദാദ്രിസംഭവത്തില്അഖ്ലാഖിന്റെവീട്ടിലിന്നുംകണ്ടെത്തിയത്ആടാണോമാംസമാ
ണെന്നറിപ്പോര്‍ട്ടിനെചോദ്യംചെയ്തഉത്തര്പ്രദേശുഖ്യമന്ത്രിഅഖിലേഷ്യാദ.
വ്.

[Chief Minister of Uttar Pradesh Akhilesh Yadav questioned on the report of Dadri incident which claimed the meat sample collected from Aqlaq's house is beef]

Different paraphrase methods based on various techniques are discussed in this section. Bill Dolan et al. constructed large paraphrase corpora from different news domains in an unsupervised way [1]. In this method, the investigation has been done on how monolingual sentence-level paraphrases are acquired using different unsupervised techniques from a corpus ranging over topical and temporal news articles brought together from thousands of news sources existing in the web. Another approach based on semantic similarity for paraphrase detection was proposed by Fernando et al. [2]. This work presented a new approach toward solving paraphrase identification problem. Distributional analysis based on linguistics for dealing paraphrases is explained in [3]. Lin et al. described the generation of semantic vector representations for sentences using recursive autoencoders [4, 5]. Kalchbrenner et al. [6] explained the idea of dynamic pooling and the importance of short- and long-range relations over sentences.

Finch et al. proposed the method to determine the sentence-level semantic equivalence using machine translation evaluation [7]. The machine translation (MT) evaluation task and sentence-level semantic similarity classification are strongly related. Socher et al. [8] explained detecting paraphrases using unfolding

recursive autoencoders. The sentence length was fixed using the dynamic pooling layer concept, and further details on this work are explored in [9]. The same algorithm implementation with slight modifications is used in this proposed system.

Amrita-CEN organized a shared task, Detecting Paraphrases for Indian Languages (DPIL) at the Forum for Information Retrieval Evaluation-2016 (FIRE-2016) [10]. The Indian languages concerned in this task are Malayalam, Hindi, Tamil, and Punjabi. The various features used by different teams include stem or lemma, POS, word overlap, stop words, synonym, cosine, etc. Mathew et al. discussed different statistical techniques, like Jaccard similarity, cosine similarity, dice similarity, word order, and word distance similarity used for the paraphrase identification task [11, 12]. Besides the statistical method, semantic methods such as UNL en-conversion process, UNL expression, and UNL graph-based similarity were used for detecting the paraphrases [13]. Paraphrase identification also can be applied to noisy social media data, which is found to be complicated. Paraphrase detection on Twitter data used existing methods such as word or n-gram overlapping, word alignment, and string matching to semantic word similarity [14]. Mahalakshmi et al. implemented recursive autoencoders for identifying paraphrases on twitter data [15]. Tamil paraphrase identification system using recursive autoencoders is explained in [16]. Low-level string and semantic and lexical features for paraphrase detection can be seen in [17].

Word similarity is derived using WordNet, which is further used in a semantic approach to identify the paraphrases. An unsupervised semantic is introduced in [18]. As similar to the question and answer application, paraphrase detection based on short answer scoring is discussed in [19]. A textual similarity metrics based on abductive networks is shown in [20]. Their inference shows improved results on using individual metrics. Nouns along with verb ambiguities in the paraphrase are handled by a fuzzy hierarchical clustering-based approach [21]. Brockett et al. explained text feature-based heuristic where the first two sentences of each document in a cluster are cross-matched with each other to determine similar sentences [22]. In the work proposed by He et al. [23], they investigated intrinsic features in different granularity, and finally, convolutional neural network for sentence embedding is used in paraphrase identification [23]. Mihalcea et al. explained the effectiveness of two corpus-oriented and six knowledge-oriented measures for text semantic similarity. Six knowledge-based measures are Leacock and Chodorow similarity, Lesk similarity, Wu and Palmer similarity, Resnik similarity, Lin similarity, and Jiang and Conrath similarity [24]. TF-KLD, which includes term frequency and KL divergence, is a term-weighting metric introduced by Ji et al. [25]. TF-KLD measures the discriminability of a feature, and it is found to yield better semantic relatedness between a pair of paraphrased sentences by the newly reweighted feature-context matrix factorization. Cheng et al. adopted Siamese architecture for paraphrase detection. An additional layer that lies on top of the compositional layer for scoring the linguistic plausibility of the produced phrase or sentence vector with regard to both syntaxes, as well as semantics, is explained in [26]. An investigation on how features based on syntactic dependencies will affect paraphrase detection is carried out in [27]. Paraphrase detection based on the significance of

dissimilarities present in sentences is explained in [28]. Linguistic methods are showing better results in identifying paraphrases than statistical methods in [29].

Identification of paraphrases in Tamil is also considered to be a complex task like that of Malayalam because both are languages that fall in Dravidian category possessing similar semantic structures and morphological variations. For the same reason, extensive researches are happening with different applications of NLP in Tamil language domain also. Some of them are discussed here.

A language independent paraphrase identification system is introduced in [30] based on statistical features such as Jaccard similarity, edit distance based on length, and cosine similarity. They proposed a probability-based neural network for the detection task, which is trained by the abovementioned statistical features. The paraphrase detection method discussed in [31] explains about using multinomial logistic regression trained with lexical level and semantic level similarities as its features. [32] discusses on a study conducted by them on existing paraphrase identification tenchniques and its application to perform paraphrase detection automatically. Sentence similarity evaluation on sentences having named entities is illustrated in [33]. A novel approach of paraphrase identification using Collaborative and Adversarial Network is introduced in [34, 35] and reveals the importance and different issues affected by robustness in identifying paraphrases. Context learning using lexical, sentential, and syntactic encodings is discussed in [36]. Praveena et al. illustrated paraphrase identification on Malayalam by learning chunking-based semantic features [37]. Different supervised techniques for paraphrase identification on Quora and Twitter data are discussed in [38]. Elaborated explanation on creation of dataset for detecting paraphrases in four different Indian languages is given in [39].

2 Materials and Methods

Let $S_1 = \{W_1, W_2. \ldots . W_m\}$ and $S_2 = \{W_1, W_2. \ldots . W_n\}$. The semantic closeness between the sentences decides to which type of paraphrase they belong to. As mentioned earlier, initial problem was to classify sentence pairs into paraphrases or non-paraphrases, which is called a two-class classification problem. The extension of this system identifies semi-equivalent paraphrases also, and it is termed a three-class classification problem. If the semantic closeness between the sentence pairs is high, they are classified as paraphrases, and if less, they are classified as non-paraphrases. Those that lie in a middle range fall into semi-equivalent paraphrases. That is,

$$P \rightarrow SC(S_1, S_2) = \text{high}$$

$$NP \rightarrow SC(S_1, S_2) = \text{low}$$

$$SP \rightarrow SC(S_1, S_2) = \text{average}$$

where SC is the semantic closeness between sentences.

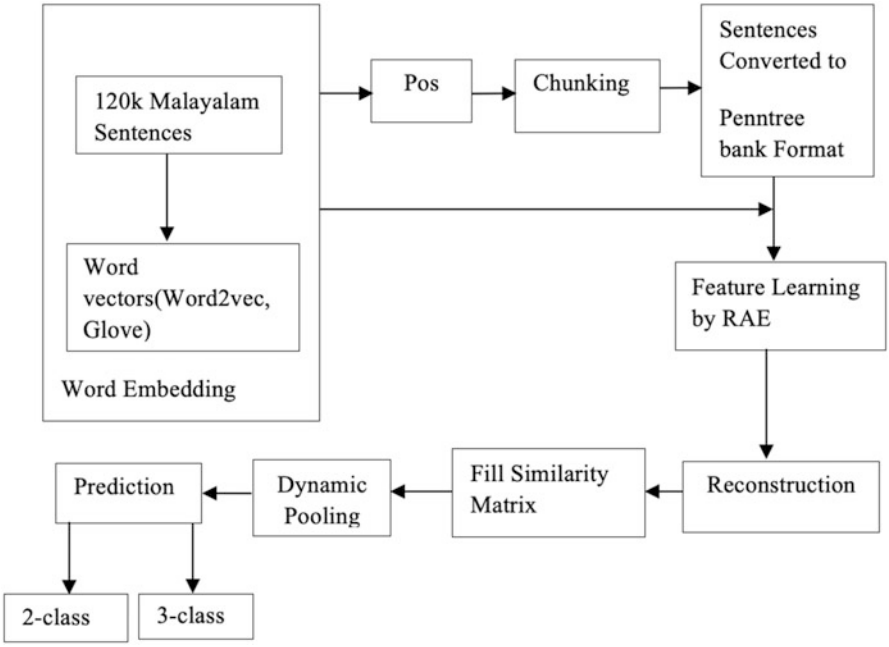

Fig. 1 Malayalam paraphrase identification system framework

2.1 Malayalam-Paraphrase Identification System

The primary experiments carried out for producing a baseline system for Malayalam are depicted in Fig. 1. The word vectors obtained from a large collection of Malayalam sentences along with supervised sentences falling in paraphrase, non-paraphrase, and semi-equivalent paraphrase categories in Penn Treebank format underwent an RNN training to generate sentence embedding. The RNN architecture used is unfolding recursive autoencoders, which are explained in Sect. 4.3. These sentences of varying lengths are altered to be in a fixed size using dynamic pooling, which is explained in Sect. 4.4. The output of dynamic pooling phase is passed to a classifier for predicting the category in which the test sentences belong to. Various procedures by which the system is made are explained in the following subsections.

2.2 Word Representations

In word representations, the given words are converted to their corresponding numeric format or representations, which are technically called vectors. This process is known as word embedding. Different word embedding techniques exist, and they

use different mathematical expressions resulting in different vectors as its outcome. In this work, we have used two well-known word embedding techniques, namely, word2vec and Glove.

2.2.1 Word2vec Embedding

Word2vec proposed by Miklov et al. is one of the most commonly used embedding techniques for most of the NLP applications [40]. Word2vec will give a distributed representation for any word in a vector space, letting learning algorithms to produce better performances in various NLP tasks. Semantic information is captured from the distributed representation of words, and those that lie closer in vector space are assumed to have high semantic closeness. Word2vec poses two embedding models, namely, skip-gram model and Continuous Bag of Words (CBOW) model, out of which the former is used in the proposed model. The skip-gram model will be predicting the surrounding words from the given center word, and mathematically, the objective of the skip-gram model is defined as

$$J = \frac{1}{t_1} \sum_{t=1}^{t_1} \sum_{-n \leqslant i \leqslant n \neq 0} logP(W_{t+i} | W_t) \tag{1}$$

This will sum the logarithmic probabilities of neighboring n words to the right as well as the left side of the target word W_t.

2.2.2 Glove Embedding

Glove is also an unsupervised algorithm for obtaining word vectors [41]. Glove produced word vectors from the co-occurrence statistics of words appearing in a huge unlabeled corpus on which training is done. The weighted least square objective will minimize the difference between word vectors and their co-occurrences. The objective function is defined as

$$J = \sum_{i,j}^{v} f\left(x_{ij}\right) \left(v_i^t \overline{v}_j + \theta_i + \overline{\theta}_j - \log x_{ij}\right)^2 \tag{2}$$

2.2.3 Parsing Using Chunking Information

Parsing phase produces a tree representation, called parse trees for all labelled sentences. The output of the parser is the sentences divided into chunks consisting of noun phrases and verb phrases. Due to the unavailability of openly available accurate Malayalam parser, all sentences are initially part-of-speech (POS) tagged.

The POS-tagged sentences got chunked using an in-house chunker. The chunked sentences are then converted into *Penn Treebank* format. Syntactic information for sentences is captured by this process of converting sentences to *Penn Treebank* format.

2.2.4 Recursive Autoencoders

Recursive autoencoders (RAE) are used for learning features from the nodes of parse trees. The aim of RAE is also extended to find the corresponding vector representations for phrases of different or varying sizes that lie in each node of parse trees. RAE uses word vectors obtained from the word embedding phase for generating vector representations of phrases, having the binary parsed tree as input in the form $P \rightarrow (\text{word } 1, \text{word } 2)$. The trees are obtained by parsing the sentences, which are explained in Sect. 4.2. Figure 2 shows how an RAE works for a sentence with four words. In the figure, PV and SV stand for phrase vector and sentence vector, respectively.

RAE produces phrase vectors in an unsupervised way. It is possible to reconstruct the parent node from child representation. For example, parent *P* is computed from its children *word* 1 *and word* 2 by a typical neural network layer using $p = f(We[c_1; c_2] + b)$ where $[c_1; c_2]$ is the concatenation of c_1 and c_2. After multiplying the parameter *We* with the concatenation of c_1 and c_2, a bias term is added. The resulting vector is applied to element-wise activation function *f* such as *tanh*. To know how effective the vectors are created, the word vectors and phrase vectors are reconstructed as in Eq. (3).

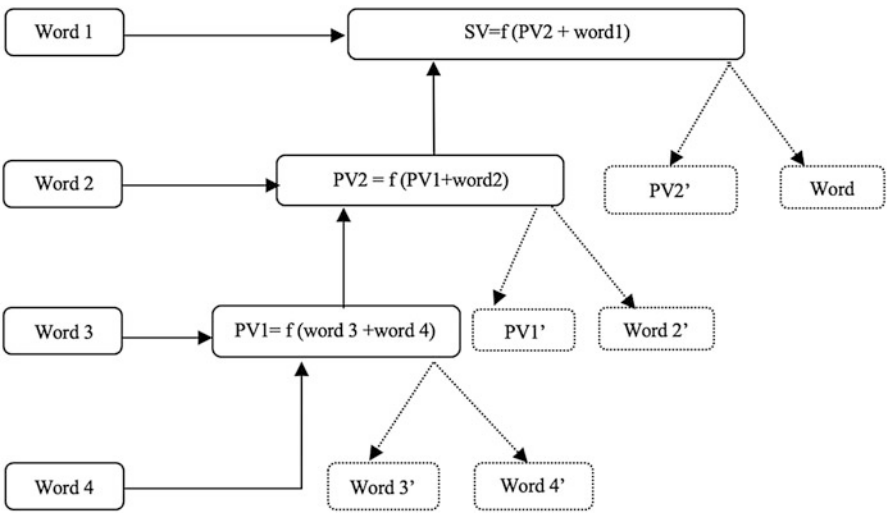

Fig. 2 Recursive autoencoders

$$\left[c_1'; c_2'\right] = We'p + b' \tag{3}$$

The Euclidean distance lies between the actual input and the reconstructed defines reconstruction error. The objective of the reconstruction phase is to have minimum reconstruction error, which is obtained using Eq. (4).

$$E_{\text{rec}}(p) = \|[c_1; c_2]\| - \left\|\left[c_1'; c_2'\right]\right\|^2 \tag{4}$$

In the proposed experiment, we used a variety of standard recursive autoencoders, called unfolding recursive autoencoders. They differ from typical RAE only in the reconstruction step. The unfolding RAE will reconstruct the entire children beneath a node when a typical RAE reconstructs only the direct children. Mathematically, the reconstruction happening in unfolding RAE is defined as in Eq. (5).

$$E_{\text{rec}}(p) = \left\|\left[x_1; \ldots\ldots; x_j\right]\right\| - \left\|\left[x_1'; \ldots\ldots; x_j'\right]\right\|^2 \tag{5}$$

We used a subset of parsed trees from the whole labeled dataset, and the sum of reconstruction error of every tree is minimized.

2.2.5 Dynamic Pooling

Sentence pairs considered for verifying paraphrase are of different sizes in terms of number of words. In order to fix the same length, we use the concept of dynamic pooling. A similarity matrix S is created from the Euclidean distance calculated for checking the reconstructed error. We know that sentences are of unequal length, and hence, the similarity matrix S also will be of unequal dimension. For converted sentences to be in a fixed length, the similarity matrix S is mapped to obtain a pooling matrix S_p, and it is divided into an approximately equal number of rows and columns. S_p contains a pooling region, and the lowest value among them is selected for further processing.

2.3 Experimental Setup

This section explains the dataset used and creation of initial word vectors for Malayalam language. It also describes the phrase vectors and the semantic based statistical features for paraphrasing.

2.3.1 Paraphrase Dataset for Malayalam

For developing an automatic paraphrase identification system in Indian languages, the paraphrased corpora are essential. We have used the Detecting Paraphrases for Indian Languages (DPIL) corpora [10] for developing a Malayalam paraphrase system. The Malayalam corpus consists of 11,000 sentence pairs that are paraphrases, non-paraphrases, and semi-equivalent paraphrases. This DPIL corpus consists of sentence pairs that are labelled using the tag "0" for non-paraphrases (NP) and 1 for paraphrases (P). This dataset was used in the two-class problem (P and NP) and the three-class problem (P, NP, and SP). The monolingual corpora consisting of 120 k Malayalam sentences were also collected from various web sources. These sentences are then used for creating the Malayalam word embedding. Table 1 describes the dataset used for paraphrase identification discussed in the proposed work. The average number of words present in source sentences and target sentences in both problems individually as well as pairwise is also described in Table 1. The sentence pairs of the proposed two-class (task 1) and three-class problem (task 2) had an average number of words as 9.253, 9.035 and 9.414, 8.449 correspondingly. The average number of words observed in a pair for task 1 is 9.144, and task 2 is 8.932.

2.3.2 Word Vector Representation for Malayalam Language

This subsection explains how the word vectors are generated for the paraphrase identification for Malayalam. Initially, 120 k randomly collected Malayalam sentences are trained to obtain vectors using word2vec [40] and Glove [41] embedding techniques. The training of unsupervised corpus gave 97,236 unique Malayalam words and their corresponding word vectors. This word embedding module results in a dictionary that consists of a sufficient number of unique words and their 100-dimensional vectors. Although the dimension of vectors was set to 100 initially, it was then extended to 200 and 300. Therefore, the size of the output file obtained after creating the word embedding is vocabulary_size × 100, vocabulary_size × 200, and vocabulary_size × 300 for the corresponding dimension.

Table 1 Dataset description for Malayalam (in pairs)

Data	Two-class	Three-class
Training	2500	3500
Testing	900	1400
Average number of words		
Sentence 1	9.253	9.414
Sentence 2	9.035	8.449
Sentence pair	9.144	8.932

2.3.3 Phrase Vector Generation and Feature Extraction for Malayalam

Recursive autoencoders (RAE) do the role of phrase vector generation. The dictionary of words and their respective words along with parsed sentences are given to RAE as input. The working of RAE is illustrated in Sect. 4.3. The different features of sentences are stored in the phrase vector generation step. After implementation of the baseline system, there are a few more statistical features based on the semantic similarity that are added to improve the system performance. They are described below.

Word Overlapping

This feature extracts the common strings appearing in a pair of the sentence. Consider sentences S_1 and S_2. Then,

$$\text{word overlapping} = S_1 \cap S_2 \tag{6}$$

Edit Distance

Consider sentences S_1 and S_2. The number of characters to be changed to produce S_1 from S_2 is the edit distance.

POS Tag

Common POS tag information in the sentences S_1 and S_2 will be added to the feature list. That is,

$$\text{POSinfo}(S_1) \cap \text{POSinfo}(S_2) \tag{7}$$

Character Length

The number of characters appearing in S_1 and S_2 gives their character length. That is,

- $\text{Cl1} = \text{length}(S_1)$.
- $\text{Cl2} = \text{length}(S_2)$.

The character length of S_1 and S_2 is also concatenated to add it as a feature. That is,

- $\text{Con\_ char\_ length} = [C_{l1};C_{l2}]$.

Word Length

The number of strings present in S_1 and S_2 gives their word length. That is,

- Wl1 = length(strsplit(S_1)).
- Wl2 = length(strsplit(S_2)).

The word length of S_1 and S_2 is also concatenated to add it as a feature. That is,

- Con_ word_ length = [W_{11};W_{12}].

3 Results

The paraphrase identification system discussed in this chapter went through a number of experiments, and the results obtained in these experiments are illustrated in this section. The initial procedure for solving this problem was to develop a two-class paraphrase identification system using embeddings, which is eventually considered as the baseline system for the two-class problem. The baseline system for the three-class system is also developed in similar fashion. The performance of the baseline system for two-class and three-class problems is illustrated in Tables 2 and 3, respectively. Since the performances of proposed systems need to be improved, a few conventional statistical features are added and explained in Sect. 5.3. While introducing each feature to the system, its performance kept on varying. Table 4 depicts the accuracy of the system when it learned with statistical features. The classification of baseline two-class task is done by linear regression. The response of the two-class and the three-class system to linear regression on adding statistical features on is depicted in Tables 4 and 5, respectively. The classification task using support vector classifiers for two-class and three-class is shown in Tables 6 and 7, respectively.

It is inferred from the above tables that the highest performance obtained for the two-class system is 83.32% using 200-dimensional word vectors acquired from word2vec embeddings. It is also found that comparatively in most of the cases, linear regression is found to classify better than support vector classifiers. Out of the

Table 2 Baseline system of two-class problem

Dimension	Word2vec	Glove
100	77.66%	77.33%
200	77.89%	77.78%
300	75.33%	75.89%

Table 3 Baseline system of three-class problem

Dimension	Word2vec	Glove
100	66.07%	65.43%
200	66.43%	65.76%
300	63.50%	64.50%

Table 4 Accuracies obtained on the addition of each new feature in two-class system using LR (all values in %)

	Word2vec			Glove		
Feature	100	200	300	100	200	300
Word overlapping	78.55	78.88	80.11	77.89	78.56	79.11
Edit distance	80.44	80.78	81.11	80.33	80.33	80.33
Common POS tag	80.56	80.95	81.00	80.56	80.22	80.63
Char count of S_1	81.44	81.56	81.56	81.00	81.11	81.67
Char count of S_2	82.22	82.44	82.11	82.56	82.33	83.22
Char count of S_1 & S_2	82.45	82.78	81.78	82.22	82.33	81.89
Word count of S_1	82.89	82.89	82.22	82.22	82.66	82.89
Word count of S_2	83.11	83.21	82.22	82.67	82.56	83.00
Word count of S_1 & S_2	83.32	83.15	82.11	82.33	82.33	82.22

Table 5 Accuracies obtained for three-class system using LR

	Word2vec			Glove		
Feature	100	200	300	100	200	300
Word overlapping	68.21	67.14	67.29	66.86	66.42	66.86
Edit distance	67.43	67.57	64.86	66.79	66.86	65.86
Common POS tag	68.29	66.50	68.64	68.36	68.57	67.00
Char count of S_1	70.93	69.85	70.79	70.71	70.57	70.36
Char count of S_2	68.00	68.07	70.21	68.57	67.93	70.79
Char count of S_1 & S_2	68.58	68.79	67.50	68.93	68.42	67.14
Word count of S_1	68.50	69.14	68.50	68.79	68.35	67.50
Word count of S_2	68.86	68.57	68.50	68.86	68.21	68.50
Word count of S_1 & S_2	69.29	69.36	68.14	69.76	69.00	68.64

Table 6 Accuracies obtained for two-class system using SVM

	Word2vec			Glove		
Feature	100	200	300	100	200	300
Word overlapping	78.11	79.00	79.66	79.00	78.78	80.22
Edit distance	79.89	80.44	81.33	80.22	80.22	80.44
Common POS tag	80.22	80.78	81.22	80.33	80.56	81.11
Char count of S_1	81.33	81.22	81.44	81.56	81.67	81.11
Char count of S_2	82.22	82.22	81.78	82.11	82.33	81.89
Char count of S_1 & S_2	82.44	82.33	82.33	82.11	82.22	81.44
Word count of S_1	82.22	82.44	82.44	82.00	82.11	82.22
Word count of S_2	82.22	82.44	82.11	82.67	82.22	82.22
Word count of S_1 & S_2	83.22	82.22	82.67	82.11	82.66	83.11

100-, 200-, and 300-dimensional vectors of word2vec embedding and Glove, 200-dimensional vectors are giving better performance compared to the other two. Graphical analysis of 100-dimensional vectors of word2vec embeddings and Glove

Table 7 Accuracies obtained on the addition of each new feature in a three-class system using SVM (all values in %)

	Word2vec			Glove		
Feature	100	200	300	100	200	300
Word overlapping	66.79	67.93	67.64	66.36	66.71	66.14
Edit distance	68.14	67.43	67.78	67.29	68.21	67.43
Common POS tag	68.14	68.36	68.86	68.50	67.93	68.14
Char count of S_1	71.07	70.64	70.50	71.43	70.79	70.58
Char count of S_2	71.50	71.29	71.27	70.93	70.93	68.64
Char count of S_1 & S_2	68.58	68.93	68.94	68.43	68.57	69.43
Word count of S_1	68.58	68.57	67.93	68.64	67.71	68.29
Word count of S_2	68.93	69.14	68.57	68.93	68.07	70.43
Word count of S_1 & S_2	69.71	69.14	69.36	68.79	68.79	69.36

Fig. 3 Graphical analysis of 100-dimensional word2vec and Glove vectors using logistic regression

using logistic regression and SVM classifier is illustrated in Figs. 3 and 4, respectively. The statistical features added with the embeddings improve the performance of the paraphrase system.

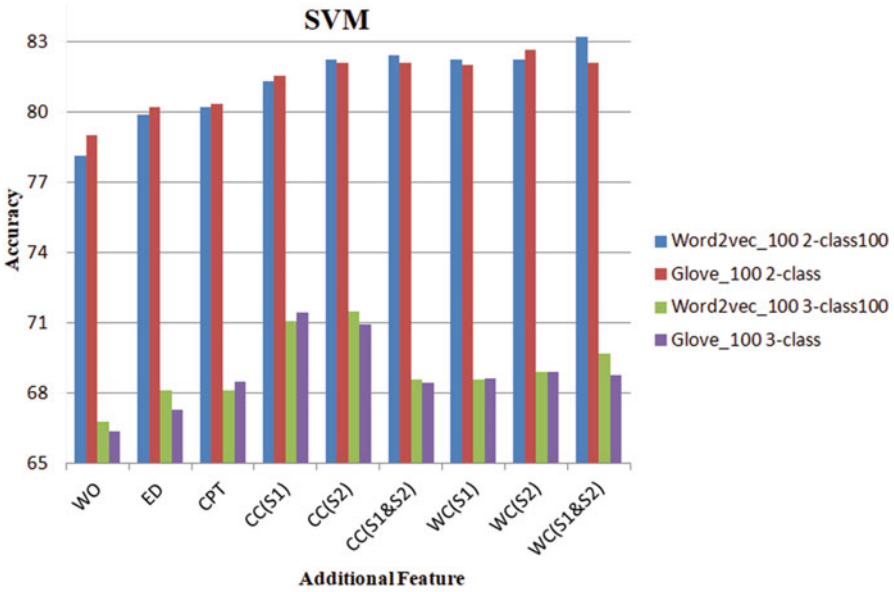

Fig. 4 Graphical analysis of 100-dimensional word2vec and Glove vectors using SVM

4 Discussion

This chapter presents paraphrase identification for the Malayalam language. The Malayalam sentences are mapped to their corresponding vectors using RAE embedding. Word2vec and Glove embeddings are used for obtaining the initial word vectors. Phrase vectors are captured from word embeddings and served as the features to the RAE. As the sentences vary in length, we apply dynamic pooling to make them into a fixed-size representation. The newly produced vectors are then passed to logistic regression and support vector machine classifier. Initially, a baseline system is set up for two-class and three-class tasks. It is later extended by adding new statistical feature sets. On adding each feature, the system performance is examined. From the results, we inferred that among word embedding techniques, word2vec performs better, and out of them, 200-dimensional vectors are giving comparatively better results. In the case of classifying methods used, logistic regression is classified more accurately than support vector classifiers. On adding more features to the baseline system, accuracy improved by approximately 5% for both two-class and three-class systems. Paraphrase identification still stays as a complex yet challenging problem in the field of Malayalam language processing. The approach used in the proposed work can disambiguate word-level ambiguities in identifying contexts for detecting paraphrases. Implementation of the same approach for other morphologically rich languages is possible soon. It helps to understand how the part-of-speech (POS) and morphological aspects of language effects in identifying paraphrases whose underlying idea is to find the semantic closeness between sentences.

References

1. B. Dolan, C. Quirk, C. Brockett, Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources, in *Proceedings of the 20th International Conference on Computational Linguistics*, (Association for Computational Linguistics, 2004), p. 350
2. S. Fernando, M. Stevenson, A semantic similarity approach to paraphrase detection, in *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, (Association for Computational Linguistics, 2008, March), pp. 45–52
3. M. Redington, N. Chater, S. Finch, Distributional information: A powerful cue for acquiring syntactic categories. Cognit. Sci. **22**(4), 425–469 (1998)
4. M. Iyyer, J. Boyd-Graber, H. DauméIII, Generating sentences from semantic vector space representations, in *Nips workshop on learning semantics*, (Semantic Scholar, 2014)
5. Lin, T.Y; And Lee, C.Y. (2013). Learning Meanings for Sentences with Recursive Autoencoders, University of California, San Diego
6. N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences. arXivpreprint arXiv **1404**, 2188 (2014)
7. A. Finch, Y.S. Hwang, E. Sumita, Using machine translation evaluation techniques to determine sentence-level semantic equivalence, in *Proceedings of the Third International Workshop on Paraphrasing*, (IWP, 2005)
8. R. Socher, E.H. Huang, J. Pennin, C.D. Manning, A.Y. Ng, Dynamic Pooling and Unfolding Recursive Auto Encoders for Paraphrase Detection, in *Advances in Neural Information Processing Systems*, (NeurIPS Proceedings, 2011), pp. 801–809
9. E. Huang, Paraphrase detection using recursive autoencoder (2011), http://nlp.stanford.edu/courses/cs224n/2011/reports/ehhuang.pdf. Accessed 21 Sept 2016
10. K.M. Anand, S. Singh, B. Kavirajan, K.P. Soman, DPIL@ FIRE2016: Overview of Shared Task on Detecting Paraphrases in Indian Languages, in *Working Notes of FIRE 2016–Forum for Information Retrieval Evaluation*, (CEUR Workshop Proceedings, Kolkata, India, 2016), pp. 7–10
11. K. Manju, S.M. Idicula, CUSAT_TEAM@DPIL-FIRE2016: detecting paraphrase in Indian languages-malayalam. In FIRE (Working Notes), 279–281 (2016)
12. L. Sindhu, S.M. Idicula, CUSAT_NLP@ DPIL-FIRE2016: Malayalam paraphrase detection. In FIRE (Working Notes), 266–269 (2016)
13. M.I. Sumam, S. Savitha, *Comparison of Statistical and Semantic Similarity Techniques for Paraphrase Identification* (IEEE, 2012)
14. V. Ngoc Phuoc An, S. Magnolini, O. Popescu, *Paraphrase Identification and Semantic Similarity in Twitter with Simple Features* (Association for Computational Linguistics, 2015)
15. M.S. Sundaram, A.K. Madasamy, S.K. Padannayil, AMRITA_CEN@SemEval-2015: Paraphrase Detection for Twitter Using Unsupervised Feature Learning with Recursive Autoencoders, in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, (Association for Computational Linguistics, 2015), pp. 45–50
16. S. Mahalakshmi, M. Anand Kumar, K.P. Soman, Paraphrase detection for Tamil language using deep learning algorithm. Int. J. Appl. Eng. Res. **10**(17), 13929–13934 (2015)
17. E. Pronoza, E. Yagunova, Low-level Features for Paraphrase Identification, in *Mexican International Conference on Artificial Intelligence*, (Springer, Cham, 2015), pp. 59–71
18. Cordeiro, J; Dias, G; Brazdil, P. (2007). A Metric for Paraphrase Detection. *In Computing in the Global Information Technology*, 2007. ICCGI 2007. International Multi-Conference on (pp. 7). IEEE
19. N. Koleva, A. Horbach, A. Palmer, S. Ostermann, M. Pinkal, Paraphrase Detection for Short Answers Coring, in *Proceedings of the Third Workshop on NLP for Computer-Assisted Language Learning*, (LiU Electronic Press, 2014), pp. 59–73
20. E.S.M. El-Alfy, R.E. Abdel-Aal, W.G. Al-Khatib, F. Alvi, Boosting paraphrase detection through textual similarity metrics with abductive networks. Appl. Soft Comput. **26**, 444–453 (2015)

21. A. Chitra, A. Rajkumar, Paraphrase extraction using fuzzy hierarchical clustering. Appl. Soft Comput. **34**, 426–437 (2015)
22. C. Brockett, W.B. Dolan, Support Vector Machines for Paraphrase Identification and Corpus Construction, in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, (Natural Language Processing Group, 2005)
23. H. He, K. Gimpel, J. Lin, Multi-perspective sentence similarity modeling with convolutional neural networks, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Association for Computational Linguistics, 2015), pp. 1576–1586
24. R. Mihalcea, C. Corley, C. Strapparava, Corpus-based and Knowledge-based Measures of Text Semantic Similarity, in *Association for the Advancement of Artificial Intelligence*, vol. 6, (Semantic Scholar, 2006), pp. 775–780
25. Y. Ji, J. Eisenstein, Discriminative Improvements to Distributional Sentence Similarity, in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (Association for Computational Linguistics, 2013), pp. 891–896
26. J. Cheng, D. Kartsaklis, Syntax-aware multi-sense word embeddings for deep compositional models of meaning. arXivpreprint **arXiv**, 1508.02354 (2015)
27. S. Wan, M. Dras, R. Dale, C. Paris, Using dependency-based features to take the 'para-farce' out of paraphrase, in *Proceedings of the Australasian Language Technology Workshop 2006*, (Academia, 2006), pp. 131–138
28. L. Qiu, M.Y. Kan, T.S. Chua, Paraphrase recognition via dissimilarity significance classification, in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, (Association for Computational Linguistics, 2006), pp. 18–26
29. M.Y.M. Chong, *A Study on Plagiarism Detection and Plagiarism Direction Identification Using Natural Language Processing Techniques* (Semantic Scholar, 2013)
30. S. Sarkar et al., NLP-NITMZ@ DPIL-FIRE2016: Language Independent Paraphrases Detection, in *FIRE (Working Notes)*, (Semantic Scholar, 2016)
31. K. Sarkar, KS_JU@ DPIL-FIRE2016: detecting paraphrases in indian languages using multinomial logistic regression model. arXiv preprint **arXiv**, 1612.08171 (2016)
32. A. Altheneyan, M.E.B. Menai, Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection. *Int. J. Patter. Recogn. Artif. Intel.* **34**(04), 2053004 (2020)
33. M. Mohamed, M. Oussalah, A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics. Lang. Resour. Eval. **54**(2), 457–485 (2020)
34. J.A. Alzubi, R. Jain, A. Kathuria, A. Khandelwal, A. Saxena, A. Singh, Paraphrase identification using collaborative adversarial networks. J. Intel. Fuzzy Syst. **39**(1), 1021–1032 (2020)
35. Z. Shi, T. Yao, J. Xu, M. Huang, Robustness to modification with shared words in paraphrase identification. *arXiv preprint **arXiv***, 1909.02560 (2019)
36. S. Xu, X. Shen, F. Fukumoto, J. Li, Y. Suzuki, H. Nishizaki, Paraphrase identification with lexical, syntactic and sentential encodings. Appl. Sci. **10**(12), 4144 (2020)
37. R. Praveena, M.A. Kumar, K.P. Soman, Chunking Based Malayalam Paraphrase Identification Using Unfolding Recursive Autoencoders, in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (IEEE, 2017), pp. 922–928
38. S. Viswanathan, N. Damodaran, A. Simon, A. George, M.A. Kumar, K.P. Soman, Detection of Duplicates in Quora and Twitter Corpus, in *Advances in Big Data and Cloud Computing*, (Springer, Singapore, 2019), pp. 519–528
39. S. Singh, P. Ramanan, V. Sinthiya, K.P. Soman, Creating Paraphrase Identification Corpus for Indian Languages: Opensource Data Set for Paraphrase Creation, in *Handbook of Research on Emerging Trends and Applications of Machine Learning*, (IGI Global, 2020), pp. 157–170
40. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space. arXivpreprint **arXiv**, 1301.3781 (2013)
41. J. Pennington, R. Socher, C. Manning, Glove: Global Vectors Forword Representation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2014), pp. 1532–1543

Model Matching: Prediction of the Influence of UML Class Diagram Parameters During Similarity Assessment Using Artificial Neural Network

Alhassan Adamu, Salisu Mamman Abdulrahman, Wan Mohd Nazmee Wan Zainoon, and Abubakar Zakari

1 Introduction

Artificial intelligence (AI) has shown a remarkable ability to learn and predict complex relationship sets of parameters. Many problems that prove to be intractable because of their complexity or sheer size have been solved using the AI approach [1]. Researchers on model matching are faced with the same problem of complexity and sheer size nature of model parameter matching. This necessitates them to employ the use of optimization algorithms such as genetic algorithm [2–5], particle swarm optimization algorithm as in the work of [6, 7], cuckoo search algorithm [8], and dynamic programming approach [9] to find near-optimal solution when computing the similarity between model parameters. UML models consist of a number of diagrams that represent the view of a software system from a different perspective. For example, class diagrams are used to represent the structural view of a software system, sequence and use case diagrams to represent the functional view of a system, and state machine diagrams to represent the behavior of a system [10].

Additionally, each diagram representing a view consists of a number of parameters or elements; for example, a class diagram consists of class names, attribute names, method names, and structural relations. Accordingly, these parameters represent different meanings in a diagram in a particular instance. Because of the nature of model elements, it becomes difficult to capture and compute the similarity between two different diagrams of models by relying on one or two parameters; this problem motivates some authors to propose different similarity measures that

A. Adamu · S. M. Abdulrahman · A. Zakari (✉)
Department of Computer Science, Kano University of Science and Technology, Kano, Nigeria

W. M. N. W. Zainoon
School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_6

can capture the similarity between model diagrams by considering different elements in a diagram as in the work of [11, 12]. However, existing works failed to show the contribution and influence of each parameter during the similarity computation. To the best of our knowledge, there is no existing work that proposed the use of an artificial intelligence approach to predict the influence of the model's parameters during the similarity assessment between them.

This chapter is organized as follows: Sect. 2 provides an overview of artificial neural network, Sect. 3 present related works of ANN in software engineering domain, Sect. 4 describes the proposed framework, Sect. 5 presents result analysis and discussion and Sect. 6 give the conclusion.

2 Artificial Neural Network (ANN)

The motivation of using ANNs in this study is that ANNs are data processing mechanisms that do not follow specific pattern when processing data but use the existing data received as input to discover/learn the rules governing them. This mechanism makes ANNs powerful in solving problems with data at hand, and we do not know how those data are related to one another. ANNs are mathematical modeling tools that are used in the prediction and forecasting of complex relations among data. It was historically designed to operate through simulating/mimicking the activity of the human brain [13], which was accomplished through a large number of highly interconnected processing elements. ANNs have three structures: the input layer, where the data are imputed to the model; the hidden layer, where the data are processed; and the output layer, where the result is produced [14]. The ANN structure is characterized by three components: neurons (nodes), weights, and activation function [15]. Figure 1 shows a simple example of neural networks, consisting of one input layer, two hidden layers, and two output layers.

Neurons (nodes) are the basic unit of ANN. They interconnect one other via links known as synapses [13], which are used to send signals to one another along with weighted connections. The nodes serve as the input to the network; they are processed to produce the output [16]. In the process, weights are adjusted in such a way that for a given input the desired output is produced based on the learning

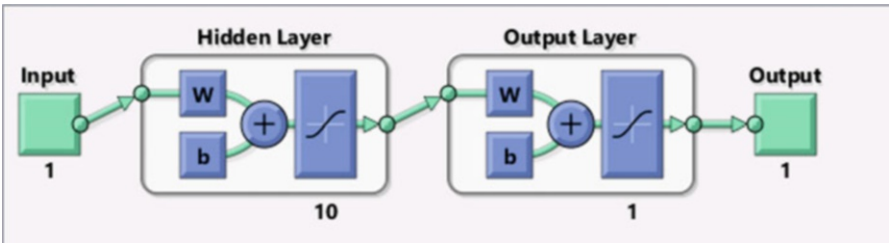

Fig. 1 Three-layer feedforward ANN

process (algorithm) and is used to reduce the error between the observed and the predicted target [17]. ANN has various classifications such as feedforward neural network (FFNN).

3 Related Works

The application of ANN is not new in the field of software engineering problems. Several problems have been tackled with ANN in various phases of software engineering right from analysis to software testing [18–20]. A review of the existing works on the application of ANN on software engineering problems [21] divides the application area into (a) software project cost estimation, (b) software metrics, (c) software testing, and (d) software quality and reliability predictions. Interested readers can refer to [21] for details. At the moment, to the best of our knowledge, there are no existing works on the application of ANN in software reuse.

3.1 ANN in Software Testing

One of the earliest works is the work of Khoshgoftaar and Szabo [22], which proposed the prediction of a number of faults in software systems. Two neural network models were trained with observed (raw) data and predictive (using principal component analysis) data. The authors compare the predictive quality of the two models using data collected from two similar systems. The results show that the prediction of faults using the neural-network model with principal component measures as input outperformed the neural network raw input.

Kanmani and Uthariaraj [23] proposed an approach to predicting software fault in object-oriented software. Two neural network-based approaches were introduced: probabilistic neural networks (PNN) and backpropagation neural network (BPN). A total of 1185 dataset was collected from graduate students' projects, out of which two-thirds of the classes (790) in the software formed the training set and one-third of the classes (395) formed the test data. The models found 317 classes with the fault in the training set and 158 classes with the fault in the test set. PNN was highly robust in predicting the five quality parameters (misclassification rates, correctness, completeness, effectiveness, efficiency, and compare the prediction accuracy).

Another fault prediction with a neural network is proposed in the work of [24]. The authors proposed a novel approach called feature selection (FS) to enhance the performance of the layered recurrent neural networks (L-RNN). L-RNN is a classification technique for solving software fault prediction problem. Wrapper feature selection algorithm based on binary genetic algorithm, binary particle swarm algorithm, and binary ant colony optimization algorithm was employed. A total of 19 real software fault projects from the repository were examined with various sizes (i.e., 109–909 instances). The data is divided into training (80%) and

testing (20%). The experiments were tested with and without feature selection; several criteria are used in evaluating the classifiers such as accuracy, precision, recall, F-measure, and area under the curve (AUC). The results show that L-RNN is able to obtain a good classification rate of average AUC of 0.8358 overall the datasets.

In the work of [25], an approach of predicting the software quality by combining multiple classifiers was proposed. The study tries to answer certain research questions, such as what is the powerful base predictor algorithm for evaluation of fault in a software system, how the number of predictors affects fault detection performance, and what is the possible ensemble combination that might change the performance of ensemble predictors. An experiment was conducted to empirically demonstrate the fault-prediction performance of ten ensemble predictors. A total of 15 software projects were evaluated for fault-detection performance of the algorithms. The study demonstrates that ensemble predictors improve the software fault prediction based on the performance measure of F-measure and area under the operating characteristics (ROC). Furthermore, Ghosh and Singh [26] proposed deep learning technique for software fault prediction using convolutional neural network (CNN). CNN is a deep learning network that reduces the number of parameters in a dataset. The model is trained with a training dataset; after the model is trained, the test data is sent for fault localization. Moreover, Serban and Bota [27] present an empirical investigation of combining two software metrics with feature selection method in predicting fault accuracy. Their experiment reveals that a combination of high-performing metrics predicts a greater number of bugs in software class with higher precision. Several other optimization and natural language processing techniques can be used [29–32].

4 Framework Overview

This section discusses the architecture of feedforward network and backpropagation training algorithm (referred to as backpropagation neural network (BPN)) for class diagram parameter prediction. Class diagrams consist of numerous parameters used in computing the similarity between models in software systems. Class depicts the structural representation of a software system. Figure 2 shows an example of a class diagram for the railway system.

Parameters in the class diagram consist of the class name, attributes, methods, and relationships. These parameters determined the relatedness between one system and another. For example, in the work of [2], the similarity between the class diagrams was computed using relationships between the class diagram classifiers. The relationships are stored in an adjacency matrix, which holds the measure of the degree of dissimilarity between the various types of relationship. Similarly, the works of [7] compute the similarity between two class diagrams using the class names; the work measures how similar two class names using name similarity (NS). NS is calculated

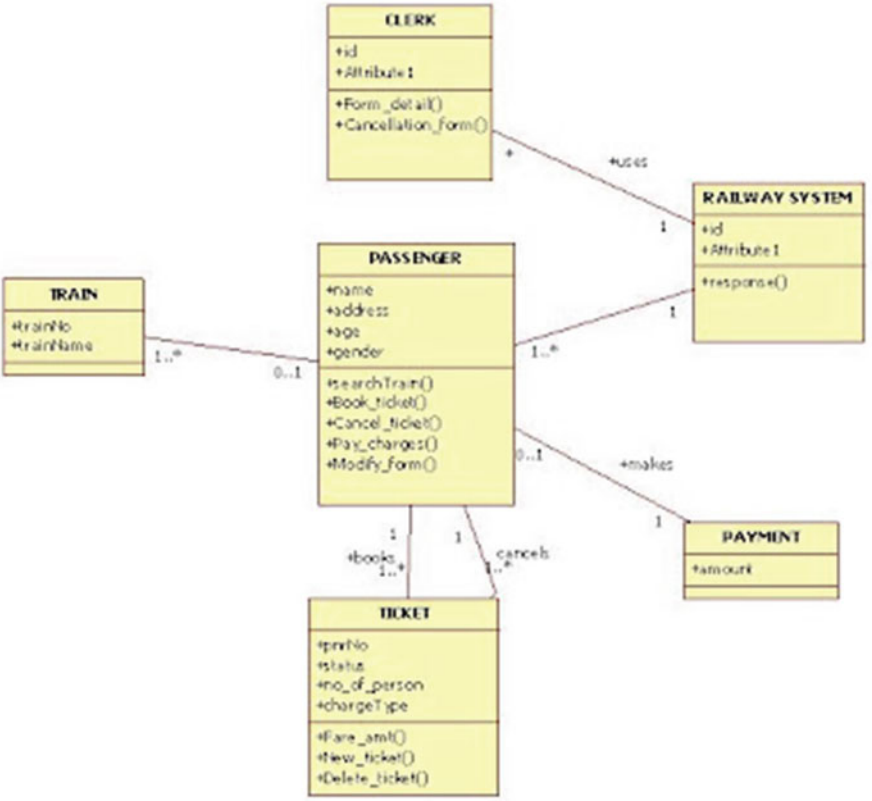

Fig. 2 Sample class diagram for the railway system

with the aid of Levenshtein distance (LD), which measures the number of characters in a string needed to be changed to obtain another character in a string.

4.1 Proposed Methodology

ANN approach is adopted in this chapter to develop model (class diagrams) parameter relationship to introduce nonlinearity occurrence rather than conventional approaches. FFBPNN contains more than one layer of neurons; with single-layered feedforward neural networks, the model has one layer of sigmoid neurons, which are then followed by an output layer of linear neurons. With sigmoid transfer functions, the model learns both linear and nonlinear relationships between input and output variable vectors. The overall view of the prediction methodology is shown in Fig. 3.

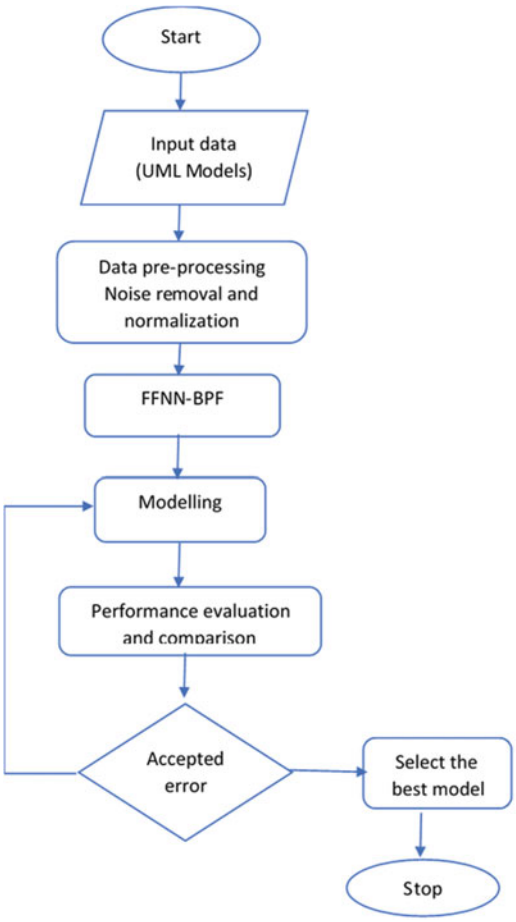

Fig. 3 Flow chart of the proposed methodology

4.1.1 Data Preparation (Input Data)

Data is obtained from existing software projects spanning different types of domains such as Java Game Maker, Plot Digitizer, OpenStego, JOrtho6 (JO) 51 Degrees, and Jcurses obtained from http://sourceforge.net. The models from the software was obtained by reverse engineering using Altova® UModel (www.altova.com). A repository was created and contained five versions of each software family making a total of 30 software projects. The similarity between values between software projects was computed to obtain the observed similarity between value systems. The observed similarity values were used as one of the input values to the ANN model. Table 1 shows the sample data.

Table 1 Sample ANN model input data

CN	CA	CM	CR	O
0.810	0.564	0.989	0.667	0.758
0.810	0.542	0.948	0.617	0.729
0.810	0.542	0.948	0.595	0.724
0.750	0.542	0.948	0.595	0.709
0.750	0.542	0.948	0.595	0.709
0.750	0.539	0.948	0.595	0.708
0.735	0.539	0.948	0.594	0.704
0.735	0.514	0.948	0.594	0.698
0.682	0.514	0.948	0.594	0.685

CN class name, *CA* class attributes, *CM* class methods, *CR* class relations, *O* observed similarities

4.1.2 Data Preprocessing

The input data of UML class diagram models are class names, class attributes, class methods, and class relationships obtained from the software models in query, and repository was normalized into a common scale of 0–1 prior to model building and analysis. The input data was normalized using the normalization equation presented in Eq. (1).

$$X_s = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

where X_s is standardized value, X_i is original value, $X_{\min}$ is the minimum value of X, and $X_{\max}$ is the maximum value of X.

4.1.3 Sensitivity Analysis

Sensitivity analysis was carried to find out the contribution of input variables over the output. The sensitivity analysis result was used in model building. Pearson product-moment coefficient of correlation was for the sensitivity analysis. The Pearson correlation equation is presented in Eq. (2).

$$r = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} \tag{2}$$

where r is the Pearson product-moment coefficient of correlation, S_{xx} is the standard deviation of variable X, S_{yy} is the standard deviation of variable Y, and S_{xy} is the standard deviation of the product of variables X and Y.

4.1.4 Model Formulation

Artificial neural network (ANN) is a tremendously fast emerging technique in nonlinear modeling due to its predictive capability and ability to learn system behavior quickly. ANN is made of parallel operating architecture consisting of input, hidden, and output layers interconnected by neurons as presented in Fig. 4. ANN is trained with the association of input and target output values by activation function of hidden neurons, and its predictive capability can be improved by adjusting connection weights of each neuron until the required performance value is reached (maximum correlation coefficient or minimum mean square error between the target and output values). The critical problem in solving complex ANN architecture is obtaining the required performance value and the numbers of hidden layers as well as neurons. There are several alternatives that are tried based on the association of input and target output to represent the ANN architecture. Figure 4 shows the proposed ANN based on feedforward with a backpropagation algorithm.

The network comprises an input layer, a hidden layer, and an output layer. The required number of neurons in the hidden layer is selected by trial and error based on the best performance value. The input layer comprises two neurons, three neurons, four neurons, and five neurons, which represent the model extracted features, and the target output layer has a single neuron of field observed similarity. The strength of each connection of neurons is referred to as weight. The sum of the inputs and their weights processing into a summation operation is given in Eq. (3).

$$\mathrm{NET}_J = \sum_{i=1}^{n} W_{ij} X_{ij}. \tag{3}$$

where W_{ij} is established weight, X_{ij} is input value, and NET_j is input to a node in layer j. In the backpropagation technique, the target output neuron quantified by a sigmoid function is given in Eq. (4).

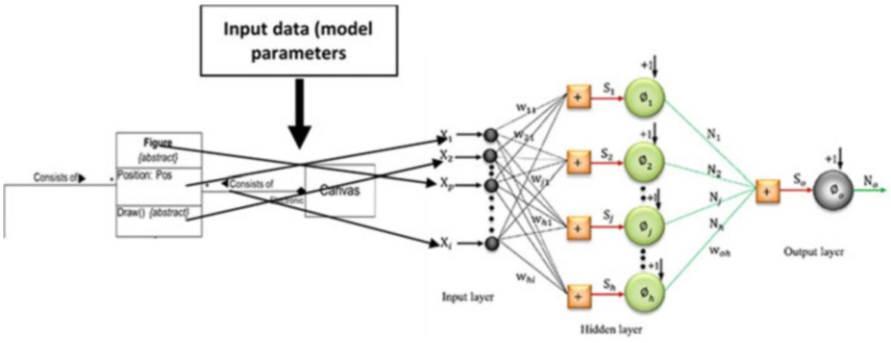

Fig. 4 Schematic structure of feedforward neural network model [33]

$$f(\mathrm{NET}_J) = \frac{1}{1 + \exp\left(-\mathrm{NET}_j\right)} \tag{4}$$

$$R$$

The backpropagation algorithm is analogous to supervised training and minimizes the sum of square error by modifying connection weights.

5 Results and Analysis

5.1 Sensitivity Analysis Results

The research uses Pearson correlation method in determining the order of importance of each variable in model building. Table 2 presents the relationship between the independent and dependent variable.

Three ANN models were build based on the Pearson correlation coefficient matrix as shown in Table 3.

5.2 Model Estimation Analysis Results

In this research, a two-layer feedforward network trained with Levenberg–Marquardt algorithm is used for the analysis of ANN models. Feedforward networks consist of a series of layers, and each subsequent layer has a connection from the previous. The model was built using a tool in MATLAB 2019a; three ANN models were developed based on the Pearson correlation coefficients. During the process, 75% of the data is used as training and 25% for validation for the analysis of ANN

Table 2 Pearson correlation coefficient matrix for model parameters

	CN	CA	CM	CR	O
CN	1				
CA	0.765301028	1			
CM	0.638018147	0.792793	1		
CR	0.708177917	0.976657	0.854106	1	
O	0.90939609	0.957195	0.817779	0.936281	1

CN class name, *CA* class attributes, *CM* class methods, *CR* class relations, *O* observed

Table 3 Model combination

Model name	Parameter combination
FFNN-M1	CA, CR
FFNN-M2	CA, CR, CN
FFNN-M3	CA, CR, CN, CM

Table 4 ANN model training phase and ANN model testing phase

	R^2	R	MSE	RMSE
Training phase				
FFNN-M1	0.981615	0.990764762	0.00005	0.00694
FFNN-M2	0.99973	0.999865146	0.00000071	0.00084
FFNN-M3	0.999968	0.999984142	0.000000083	0.000288
Testing phase				
FFNN-M1	0.97341201	0.986616	0.000105711	0.01028159
FFNN-M2	0.9999834	0.999992	0.0000001	0.00025687
FFNN-M3	0.99998718	0.999994	0.000000051	0.0002258

models. Network performance was measured according to the mean squared error (MSE). Table 4 presents the performance measure for all the three ANN models in training and validation in both ascending and descending directions.

6 Model Validations

Validation is an essential part of modeling as it demonstrates how reasonable the model represents the actual system. The most common index for evaluating the performance of ANN models are coefficient of correlation, coefficient of determination, MSE and RMSE. RMSE represents the sample standard deviation of the differences between predicted values and observed values [34]. The values of R^2, R, MSE, and RMSE are estimated using Eqs. (5)–(8).

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (O_i - P_i)^2}{\sum_{i=1}^{n} (O_i - \overline{O})} \tag{5}$$

$$R = \sqrt{R^2} \tag{6}$$

$$\text{MSE} = \frac{\sum_{i=1}^{n} (Oi - Pi)^2}{N} \tag{7}$$

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{8}$$

A total of 301 data models were collected from 30 different projects; the models consist of class diagrams with class names, and class relations, class attributes, and class methods were used for the analysis. Figure 5 shows the predictive models compared in a radar chart. The chart shows the high or low correlation of each model

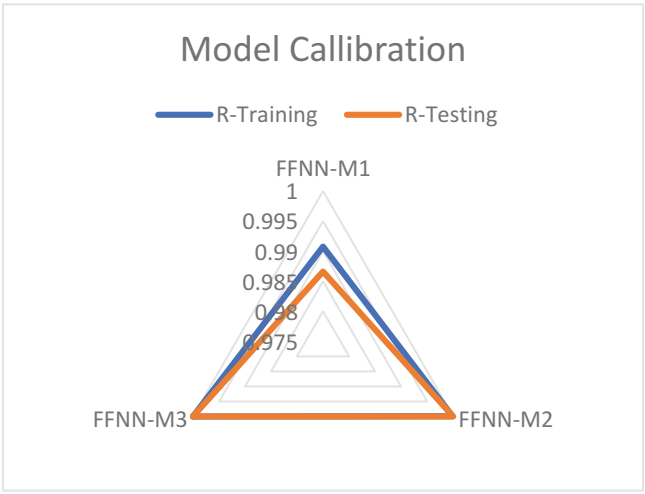

Fig. 5 Radar chart for CC in both training phase and testing phase for FFNN-M1–3

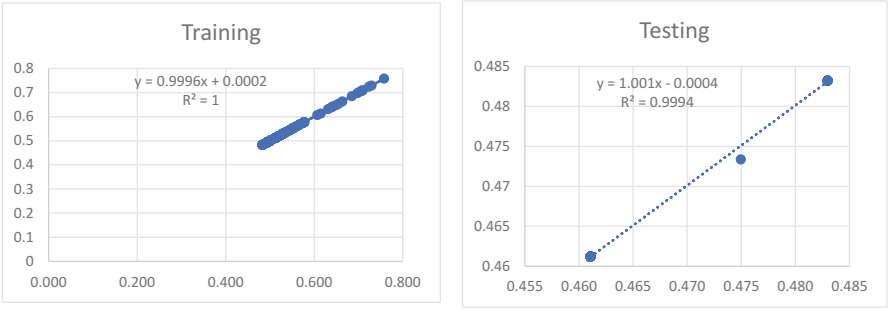

Fig. 6 Scatter plots of observed and computed values for the best model of FFN-M2

combination in order to perfectly display the performance of the model in terms of the correlation coefficient. It can be observed that 0.98 and 0.99 are both the lowest and highest value of correlation coefficient (CC) obtained from the models in the testing phase. The best performing model is attributed to the one with the high value of CC. Figure 5 shows the radar chart.

Figure 6 shows scatter diagrams of the best computed model. The plots indicate a closeness agreement between the observed and computed values for FFN-M1, FFN-M2, and FNN-M3.

7 Conclusion

This chapter presents a two-layer feedforward neural network (FFNN) trained with the Levenberg–Marquard algorithm in the analysis of ANN models to predict the influence of UML class parameters (class names, methods, attributes, etc.) when computing the similarities between software systems during software reuse. Three models were built with the aid of Pearson correlation coefficients. Each model combines a number of parameters that influence the computation of similarities between software designs. During the process, 75% of the data was used as the training and 25% for validation. The performance of the model was measured using the mean squared error (MSE). Model 2 (FFNN-M2) is the best model having a good correlation coefficient with only a few parameters as input to the ANN models.

References

1. A.S. Tenney et al., Application of artificial neural networks to stochastic estimation and jet noise modeling. AIAA J., 1–12 (2020)
2. H.O. Salami, M. Ahmed, Class Diagram Retrieval Using Genetic Algorithm, in *Machine Learning and Applications (ICMLA), 2013 12th International Conference*, (IEEE, 2013)
3. H.O. Salami, M. Ahmed, Retrieving Sequence Diagrams Using Genetic Algorithm, in *Computer Science and Software Engineering (JCSSE), 2014 11th International Joint Conference*, (IEEE, 2014)
4. H.O. Salami, M. Ahmed, A framework for reuse of multi-view UML artifacts. arXiv preprint **arXiv**, 1402.0160 (2014)
5. H.O. Salami, M.A. Ahmed, *A Framework for Class Diagram Retrieval Using Genetic Algorithm* (SEKE, 2012)
6. G. Assuncao, W. Klewerton, S.R. Vergilio, A Multi-objective Solution for Retrieving Class Diagrams, in *Intelligent Systems (BRACIS), 2013 Brazilian Conference*, (IEEE)
7. W.K.G. Assunçao, S.R. Vergilio, Class Diagram Retrieval with Particle Swarm Optimization, in *The 25th International Conference on Software Engineering and Knowledge Engineering*, (SEKE, 2013)
8. A. Adamu, W.M.N.W. Zainon, Matching and Retrieval of State Machine Diagrams from Software Repositories Using Cuckoo Search Algorithm, in *International Conference on Information Technology*, (IEEE, Al Zaytoonah University of Jordan, Amman, Jordan, 2017)
9. A. Adamu, W.M.N.W. Zainon, Similarity Assessment of UML Sequence Diagrams Using Dynamic Programming, in *International Visual Informatics Conference*, (Springer, 2017)
10. M. Ahmed, Towards the Development of Integrated Reuse Environments for UML Artifacts, in *ICSEA 2011, The Sixth International Conference on Software Engineering Advances*, (Semantic Scholar, 2011)
11. M.A.-R. Al-Khiaty, M. Ahmed, UML class diagrams: Similarity aspects and matching. Lect. Notes Softw. Eng. **4**(1), 41–47 (2016)
12. Adamu, A. and W.M.N.W. Zainoon, Determining the similarity of UML-models by combining different software properties. J. Theor. Appl. Inf. Technol., 2018. 96(11) 1992-8645
13. M.M. Hamed, M.G. Khalafallah, E.A. Hassanien, Prediction of wastewater treatment plant performance using artificial neural networks. Environ. Model Softw. **19**(10), 919–928 (2004)
14. K.P. Singh et al., Artificial neural network modeling of the river water quality—A case study. Ecol. Model. **220**(6), 888–895 (2009)

15. E. Dogan, B. Sengorur, R. Koklu, Modeling biological oxygen demand of the Melen River in Turkey using an artificial neural network technique. J. Environ. Manag. **90**(2), 1229–1235 (2009)
16. E. Dogan et al., Application of artificial neural networks to estimate wastewater treatment plant inlet biochemical oxygen demand. Environ. Prog. **27**(4), 439–446 (2008)
17. S.I. Abba, G. Elkiran, Effluent prediction of chemical oxygen demand from the wastewater treatment plant using artificial neural network application. Proc. Comput. Sci. **120**, 156–163 (2017)
18. L. Abualigah et al., Advances in meta-heuristic optimization algorithms in big data text clustering. Electronics **10**(2), 101 (2021)
19. L. Abualigah et al., Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis. Algorithms **13**(12), 345 (2020)
20. L.M.Q. Abualigah, *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering* (Springer, Switzerland, 2019)
21. Y. Singh et al., Application of Neural Networks in Software Engineering: A Review, in *International Conference on Information Systems, Technology and Management*, (Springer, Berlin, Heidelberg, 2009)
22. T.M. Khoshgoftaar, R.M. Szabo, Using neural networks to predict software faults during testing. IEEE Trans. Reliab. **45**(3), 456–462 (1996)
23. S. Kanmani et al., Object-oriented software fault prediction using neural networks. Inf. Softw. Technol. **49**(5), 483–492 (2007)
24. H. Turabieh, M. Mafarja, X. Li, Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. Expert Syst. Appl. **122**, 27–42 (2019)
25. F. Yucalar et al., Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability. Eng. Sci. Technol. Int. J. **23**(4), 938–950 (2020)
26. D. Ghosh, J. Singh, *A Novel Approach of Software Fault Prediction Using Deep Learning Technique*, in *Automated Software Engineering: A Deep Learning-Based Approach*, (Springer, Switzerland, 2020), pp. 73–91
27. C. Serban, F. Bota, A Conceptual Framework for Software Fault Prediction Using Neural Networks, in *International Conference on Modelling and Development of Intelligent Systems*, (Springer, Switzerland, 2019)
28. S.R. Chidamber, C.F. Kemerer, A metrics suite for object oriented design. IEEE Trans. Softw. Eng. **20**(6), 476–493 (1994)
29. L.M. Abualigah et al., Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Expert Syst. Appl. **84**, 24–36 (2017)
30. L. Abualigah, A. Diabat, *Advances in sine cosine algorithm: A comprehensive survey*. Artif. Intell. Rev. **54**, 2567–2608
31. L. Abualigah et al., *The arithmetic optimization algorithm*. Comput. Meth. Appl. Mech. Eng. **376**, 113609 (2021)
32. L. Abualigah, A. Diabat, A comprehensive survey of the grasshopper optimization algorithm: Results, variants, *and applications*. Neural Comput. Appl., 1–24 (2020)
33. V. Nourani et al., An emotional artificial neural network for prediction of vehicular traffic noise. Sci. Total Environ. **707**, 136134 (2020)
34. V. Nourani, T. Khanghah, A.H. Baghanam, Application of entropy concept for input selection of wavelet-ANN based rainfall-runoff modeling. J. Environ. Inf. **26**(1), 52–70 (2015)

Classical and Deep Learning Data Processing Techniques for Speech and Speaker Recognitions

Aakshi Mittal, Mohit Dua, and Shelza Dua

1 Introduction

Speech is a human behavioral trait that involves the characteristics that can distinguish the individuals and present the matter of saying of utterance. It is becoming a very low-cost and healthy way of interaction with artificially intelligent systems. Various examples of speech-driven systems can be seen these days, such as YouTube closed captioning, Amazon Alexa, speech-driven biometrics and locking systems, etc. These systems are mostly influenced with either speech recognition or speaker recognition.

Speech recognition technique emulates an utterance with its corresponding text. Speech signal involves information of context in itself, which can be achieved from spoken utterance in the form of speech feature vector [1]. To perform speech recognition, feature vector of a particular utterance is linked with its corresponding expected phonetic [2, 3].

Speaker recognition is a way of identifying individuals by recognizing their voices. Two individuals sound different due to dissimilarity in their voice production systems and their manner of presenting the utterance [4]. These physical and characteristic traits are involved in the utterance of an individual. In the case of speaker recognition, these traits become speaker-specific information that can be figured out with the help of speech feature extraction.

A. Mittal (✉) · M. Dua · S. Dua
Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana, India

Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, Haryana, India
e-mail: er.mohitdua@nitkkr.ac.in

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_7

Speech and speaker recognition systems have two parts, which are front end and back end. Front-end part of the systems uses a speech feature extraction technique that provides a feature vector. There are various classical feature extraction techniques like Mel Frequency Cepstral Coefficients (MFCC), Gammatone Frequency Cepstral Coefficients (GFCC), Perceptual Linear Prediction (PLP), etc., and deep learning-based feature extraction techniques [5–7]. Integrated features are also used to cope with the noisy environmental conditions [8, 9]. This chapter discusses the various classical and deep learning techniques and presents the analysis of front-end features used in speech and speaker recognitions. This chapter provides the implementation details for extraction of MFCC features. For the back end, a classification model is required, which is generally a machine learning model. Hidden Markov models, Gaussian mixture model, support vector machine, etc., are the widely used classical machine learning models. However, deep learning models like convolutional neural networks, long short-term memory networks, etc., are also playing a better role in performance improvement of these systems [10–12].

Various speech corpora are available to do the robust training of these systems. These corpora have speech data, related text data, knowledge of speaker, meta data for audio and text, etc. Speech-based systems are affected with the presence of noise in the environment; for example, when speech signal travels though the wireless network, it faces different types of interferences. To cope with the presence of noise, speech-based systems are made adaptable to noise by including noisy data in the training [9]. This chapter discusses various available sources of data.

This chapter is organized as follows: Sect. 2 describes private and public sources of data. Section 3 discusses various feature extraction techniques, analysis of front ends of speech, and speaker recognition techniques under clean and noisy data, discusses implementation details of static and dynamic MFCC, and discusses various deep learning-based feature extraction techniques. Section 4 concludes the presented work.

2 Sources of Data

A dataset is required for doing research and development of speech recognition and speaker recognition systems. These days, speech is easily available via some private and public sources. For the development of robust systems, variations in dataset are necessary, which are introduced by the sources unintentionally in case of private and intentionally in case of public. Some private and public sources of data are discussed below.

2.1 Private Sources of Data

Almost every organization is generating a huge amount of speech data daily. Most of the times this is the real-life data; hence, it is useful to work with. Hence, it becomes useful for research and practical works. This data can include a lot of variations in speech due to its collection from random people. This huge amount of data can be used to do research for recognition of speech or speaker. This data is private to these organizations. Sometimes, these data can include the privacy concerns of their customers and, hence, cannot be released for public use. These data can be made available for research after accomplishment of some undertakings. Some examples of these data sources are as follows:

- Telecommunication organizations
- Customer care service providers
- Television and radio data, etc.

2.2 Public Sources of Data

Research communities have developed various speech corpora to promote the research in the area of speech-driven systems. These datasets include voices of various speakers along with the meta data for the utterance. These corpora are recorded under various environmental conditions to include real-life situations in the datasets. These corpora have been recorded in different languages whether it is a country-wide language or local to any specific area. These datasets are made publically available by these communities to support research from every part of the world in speech-based areas. The following are some publically available speech corpora:

- Wall Street Journal (WSJ)
- Texas Instruments-Massachusetts Institute of Technology (TIMIT)
- National Institute of Standards and Technology (NIST)
- YOHO
- VoxCeleb, etc.

3 Data Processing

Data processing is the essential task for front-end development of any of the speech recognition or speaker recognition-based systems. Dataset provides the data in the form of audio signals, and the data processing is the way to extract the information of "What is the context of the utterance?" in case of speech recognition and "Who is saying the utterance?" in case of speaker recognition tasks. It provides information

about the context of speech and speaker-specific information in the form of numerical values that are handy for humans and different machine learning (ML) back-end models. Back-end model learns the classification clues from these coefficients and makes the prediction for the test sample. Deep knowledge of speech signal and different signal processing operations can provide the robustness to the front end of speech-driven systems. New advances in the front-end design can also be added by the application of different technologies at the front-end data processing techniques. Along with the coefficients in numeric values, there are methods of waveform and spectrogram plotting that provide the better visualization of the data.

This section discusses the various available platforms for processing the speech data. Classical speech feature extraction techniques are discussed for the speech and speaker recognition tasks. Then deep learning (DL)-based feature extraction techniques are presented in the latter part.

3.1 Available Platforms to Process the Data

Speech signal or audio from dataset can be processed with the help of various signal processing libraries provided by different languages on different software platforms. MATLAB, Python, etc., can process the speech data easily. Implementation steps discussed in this chapter are implemented in Python. There are various platforms for using Python; some are discussed below.

3.1.1 On Personal Computer

A personal computer usually provides the power of CPU and can be used to process the small amount of data. There are various integrated development environments (IDE) available to work with different languages. Anaconda is a good IDE that is supported by Windows and Linux operating systems. It provides Jupyter Notebooks, Spyder, and Qt Console platforms with good graphical user interfaces to work in Python.

This is the URL to get the Anacoondat IDE downloaded: https://docs.anaconda.com/anaconda/install/windows/.

3.1.2 On Cloud Supports

If the data to be processed is huge, then it requires the use of graphics processing unit (GPU) or tensor processing unit (TPU). These processing powers are provided by various publically available cloud supports. These cloud supports provide a fixed large unit of memory also on free of cost basis and more than that can also be acquired with some cost by contacting the relevent organization. These platforms are

highly suitable for training the machine learning (ML) models with huge dataset in less time. And these clouds keep user's work safe. The following are some of the freely available cloud supports:

- Google Colaboratory
- Kaggle Kernels
- Microsoft Azure Notebooks, etc.

3.2 Classical Feature Extraction Techniques

This section discusses the speech signal processing operations with their implementation details. Then analysis of classical feature extraction techniques is done for speech recognition and speaker recognition tasks. Implementation details of Mel Frequency Cepstral Coefficients (MFCC) with delta and delta-delta coefficients are also presented in the latter part.

3.2.1 Speech Signal Processing Operations

All the signal processing operations listed in this section are applied on *audio_file.wav* file of audio compressed in *wav* format. Any line followed by the symbol *#* represents a comment line. Python's library *librosa* is used to load, display the waveform, and process the audio. After loading the audio file by *librosa*, it provides time series *y* that is a *numpy* array and sampling rate *sr*. Along with this, *matplotlib* library is used to plot the graphs of this work (code 1). Figure 1 shows the result of the code.

Code 1

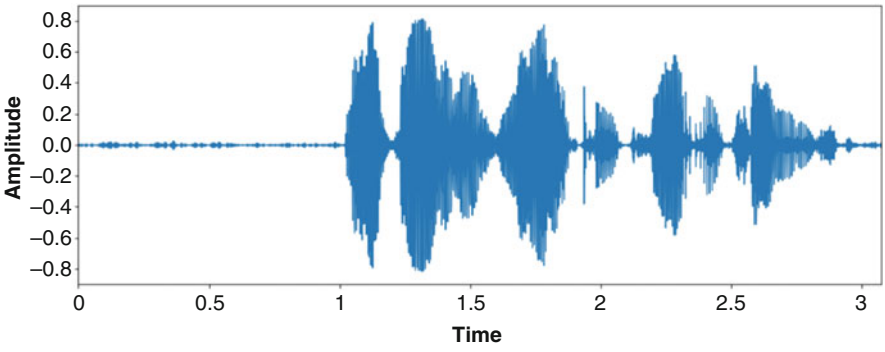

Fig. 1 Waveform of audio_file.wav

```
# Libraries
import librosa
%matplotlib inline
import matplotlib.pyplot as plt
# Whole Process
y, sr= librosa.load("audio_file.wav")
# Plotting the waveform and Figure 1 shows the output
plt.figure(figsize=(14,5))
librosa.display.waveplot(y=y, sr=sr)
plt.xlabel("Time")
plt.ylabel("Amplitude")
plt.show()
# Results
```

Numpy library provides all facilities to work with the arrays and matrices. To apply scientific and mathematical operations, *scipy* library is used, which is a dependent of *numpy* library. Both of these libraries help in applying log, square, and absolute operations on the *audio_file.wav* file. Along with these, discrete cosine transform (DCT), fast Fourier transform (FFT), and short-term Fourier transform (STFT) are also applied on the signal. These operations and transformations are applied on *y*, and hence, the results are also *numpy* arrays of values. As the results are quite large arrays, only a few columns of only the first row are shown in the results of code 2. Comment line *# More Libraries* indicates the extra libraries required, and libraries used in all previous codes are inclusive for this code too.

Code 2

```
# More Libraries
import numpy as np
from scipy.fftpack import fft, dct
 # Whole Process
# Log
logg= np.log(y)
print("Log")
print(logg)
# Square
sq= np.square(y)
print("Squares")
print(sq)
# Absolute of values
abso=np.abs(y)
print("Absolute")
print(abso)
# DCT
dctt=dct(y)
print("DCT")
print(dctt)
# FFT
fftt=fft(y)
print("FFT")
print(fftt)
# STFT
```

```
x=librosa.stft(y)
print("STFT")
print(x)
#Results
Log
[-6.0601254 -5.5124846 -5.4277854 ... ]
Squares
[5.4480606e-06 1.6289841e-05 1.9296805e-05 ... ]
Absolute
[0.00233411 0.00403607 0.00439281 ... ]
DCT
[4.9195433e+00 2.2777748e-01 3.4334455e+00 ... ]
FFT
[2.4597836+0.j 1.7166903+0.82551146j 2.526172 +1.4188728j ... ]
STFT
[4.0432158e+00+0.0000000e+00j 2.2287924e+00+0.0000000e+00j
 -1.1482446e+00+0.0000000e+00j ... ]
```

3.2.2 Analysis of Classical Feature Extraction Techniques

This section gives the detailed analysis of different feature extraction techniques in speech and speaker recognition fields. Speech recognition systems are built keeping the noisy and non-noisy environments. Below is the analysis of front ends in both of these areas.

In Speech Recognition

A lot of research has been performed for the system under ideal conditions. These researches consider the clean data for development of the systems. But when data enters into the widely spread network, it gets noisy [13]. Those particular conditions are also addressed by continuously examining the different sets of feature extraction techniques at the front end. This section discusses both clean data and noisy data scenarios.

- *With Clean Data*

Ideally, a system is considered to be situated in a noise-free environment. Therefore, a high accuracy is desired to be achieved on the clean or noiseless data. Different selections of feature extraction techniques lead the research toward better performance. Ittichaichareon et al. chose MFCC as the front-end features to train support vector machine (SVM) and maximum likelihood (ML) classifier. Dua et al. [3] trained a hidden Markov model (HMM) with heterogeneous feature vector. These heterogeneous features are generated by combining the streams of MFCC and perceptual linear prediction (PLP), which are reduced up to 39 features with the help of heteroscedastic linear discriminant analysis (HLDA) and linear transformation method. Analysis shows that heterogeneous features outperform the plane

MFCC and plane PLP features on the Hindi speech and text corpus developed by Tata Institute of Fundamental Research (TIFR), India. Elharati et al. also chose MFCC features at front end to develop Arabic speech recognition system. Dataset for this system is collected from 19 Arabic native speakers and 24 Arabic words.

- *With Noisy Data*

Speech recognition systems have gained a quite good accuracy over the years; however, their performance degrades in the presence of environmental noise and channel distortions. To make the system practiced in these difficult situations, different scenarios of noise robust features are chosen, and noisy data is generated to train the system. Këpuska et al. [14] created different hybrid features with the different combinations of MFCC, PLP, RASTS-PLT, and linear prediction coding coefficient (LPCC). A multivariate HMM is trained with these hybrid features as well as individual features under different signal-to-noise ratio (SNR). Analysis shows that static and dynamic LPCC features outperform under the high noise ratio. Dua et al. [9] used MFCC, GFCC, and basilar-membrane frequency-band cepstral coefficient (BFCC) features generated with the help of optimized filters. Applied filter banks are optimized by differential evaluation method. In this scenario, BFCC features are outperforming the MFCC and GFCC for noisy data. Dua et al. [8] used integrated features at front end and refined HMM at back end with Hindi language-based speech dataset. Integrated features are integration of MFCC + PLP and integration of MFCC + Gammatone Frequency Cepstral Coefficient (GFCC). GFCC features simulate the human auditory system and are the key point features for noise robustness. Refined back models are HMM with genetic algorithm (GA) and HMM with particle swarm optimization (PSO). Results show that MFCC + GFCC with HMM + PSO outperforms the other combinations for increased SNR (noisy data).

In Speaker Recognition

Research for speaker recognition has been done for both clean and noisy data. As this technique is applied in forensics, wireless remote access systems, etc., dealing with noisy data becomes essential [15–17]. This section highlights front-end feature extraction techniques chosen for the development of speaker recognition systems under clean and noisy environments.

- *With Clean Data*

Tiwari [18] proposed a text-dependent speaker recognition system with MFCC at front end. She analyzed the performance of the system by varying the number of filters and types of window, which shows that 32 number of filters with Hanning type of window results maximum the efficiency. Her research work shows that using less or more than 32 filters during implementation degraded the performance of the system. Sarkar et al. [19] designed a speaker recognition system specifically for

Bengali language. Their system uses different nonlinear features and is classified with the help of cross-correlation matrix. Bouziane et al. [20] presented a comparison on different feature extraction techniques for speaker recognition systems, which could enlighten the wide use of MFCC and GFCC features.

- *With Noisy Data*

Frankle and Ramachandran [21] examined speaker recognition with the presence of white noise at the level of SNR. Their research work includes five different features, namely, adaptive component weighted (ACW) cepstrum, postfilter (PFL) cepstrum, MFCC, linear predictive cepstrum (CEP), and line spectral frequencies (LSF) individually and in fused vector with the Gaussian mixture model-universal background model (GMM-UBM) at back end under varying noise scenarios. Their fusion outperforms under high noise conditions also. Bharath et al. [22] used extreme learning machine (ELM) for classification of data in noisy environment [23]. To cope with noisy environmental conditions, they used score level fusion of multiplier-based MFCC and power-normalized cepstral coefficient (PNCC) features. Desired features are obtained by the normalization of multiplier-based MFCC and PNCC. This technique performs better than the other scenarios of features and classification models for both TIMIT and SITW datasets.

3.2.3 Discussion

- In classical feature extraction techniques, MFCC is the most popular and widely used feature extraction.
- GFCC is the newly introduced feature extraction technique that is performing better than MFCC and becoming the new popular.
- Combinations or integrations of different feature extraction techniques can achieve better accuracy in noisy environments also.
- There are some features of cepstral domain like Constant Q Cepstral Coefficient (CQCC) that have not been explored in the fields of speech and speaker recognition.
- Speech and speaker recognition systems have been explored with various widely used languages of different continents, local area languages, etc.

MFCC are the most popular features in both of the discussed speech-based areas. These features are widely used individually as well as in combination of other features at the front ends of the systems of these areas. As researchers are so much inspired with these features, this chapter provides a detailed discussion on the MFCC feature extraction process with the implementation technique in the next section.

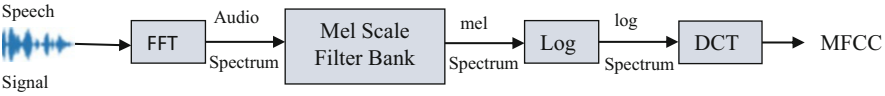

Fig. 2 MFCC feature extraction process

3.2.4 Mel Frequency Cepstral Coefficient (MFCC) Process

Mel Frequency Cepstral Coefficient (MFCC) features can process the human auditory system very well. For this, mel scale that is perceptually motivated is used. Application of FFT [24] or discrete Fourier transform (DFT) [25] on input audio provides the audio spectrum. A triangular or Gaussian shaped filter is applied to change the scale to mel scale. Final MFCC are achieved by the application of DFT to the spectrum, which is preceded by the logarithm operation [26]. Figure 2 depicts the whole process of MFCC extraction. Generally, 12 to 14 feature coefficients are reliable for different systems. The mathematical process of MFCC extraction is as follows:

$$D_{\mathrm{DFT}}(a) = \mathrm{DFT}(f) \tag{1}$$

$$\mathrm{MFB}(p) = \sum_{a=1}^{L} |D_{\mathrm{DFT}}(a)|^2 W(a) \tag{2}$$

$$\mathrm{MFCC}(r) = \sum_{p=1}^{P} log\left[\mathrm{MFB}(p)\right] \cos\left\{\frac{r(p-0.5)\pi}{P}\right\} \tag{3}$$

where f is the audio frame for which $D_{\mathrm{DFT}}(a)$ is its DFT, MFB(p) is the mel scaled frequency spectrum calculated by a^{th} mel filter bank $W(a)$ having P number of filter banks, L represents the DFT indices in total, and r MFCC features are carried out by MFCC(r).

Now, this section discusses the process of Mel Frequency Cepstral Coefficient (MFCC) extraction with implementation details. At first, FFT is applied to the audio signal before applying the mel filter to the audio. Then log operation is applied that is followed by the DCT. Code 3 uses *python_speech_features* library that facilitates the *mel* filter for audio. Again the results are huge *numpy* arrays; hence, only a few values of first row are shown in results.

Code 3

```
# More Libraries
import python_speech_features
# Whole Process
# FFT
fft=fft(y)
print("FFT")
print(fft)
```

```
# mel filter
mfc=python_speech_features.base.hz2mel(fft)
print("mel coefficients")
print(mfc)
#Log
logg=np.log(mfc)
print("After Log")
print(logg)
# DCT
dctt=dct(logg)
print("After DCT final MFCC")
print(dctt)
# Results
FFT
[2.4597836+0.j 1.7166903+0.82551146j 2.526172 +1.4188728j ... ]
mel coefficients
[3.9532394+0.j 2.7612073+1.3258146j 4.0620995+2.2761562j ... ]
After Log
[1.3745353+0.j 1.1193992+0.44764802j 1.5382305+0.510747j ... ]
After DCT final MFCC
[-8.4915918e+03-5.5313110e-05j 5.9848289e+00-1.2086787e+04j
3.2515691e+05+1.0052490e+00j ... ]
```

Code 3 teaches the whole process of MFCC extraction. As the last line of the results shows that the values of generated coefficients belong to the very large range of continuous domain, a coefficient normalization process is required before further use of these coefficients. There are quite good operations provided by the statistics to do so. MFCC can be calculated by using inbuilt function of *librosa*. In Figs. 3, 4 and 5, spectrographic views of these features are shown. Code 4 extracts the 12 MFCC features from *audio_file.wav*, which can be extracted in desired number by setting the value of *n_mfcc* parameter in the function. Dynamic feature coefficients of first and second order are also extracted in this code, and spectrograms are also plotted for them. Dynamic features are helpful in revealing the information of context of the

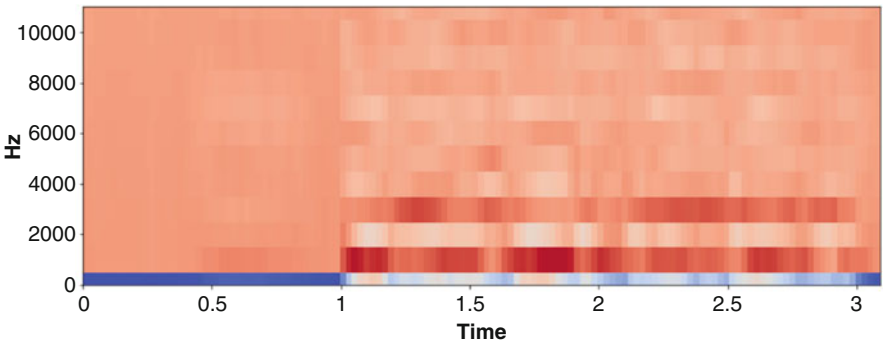

Fig. 3 Spectrographic view of MFCC features

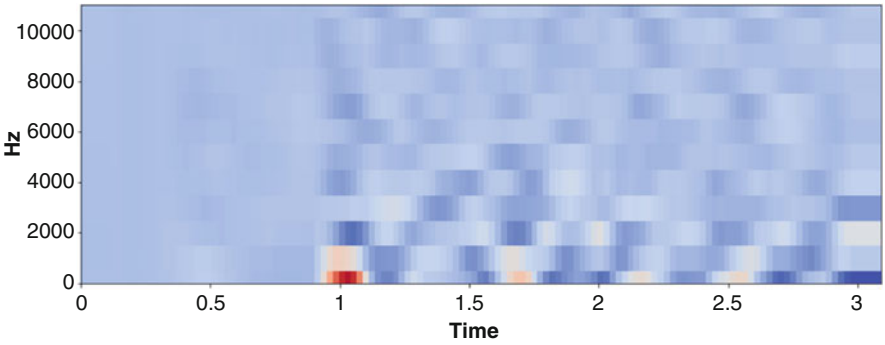

Fig. 4 Spectrographic view of delta of MFCC features

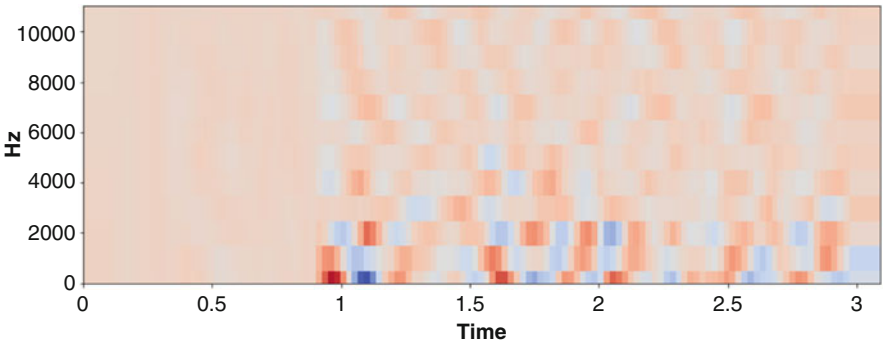

Fig. 5 Spectrographic view of delta-delta of MFCC features

utterance more clearly than only MFCC features. Figures 3, 4, and 5 represent the output of code 4.

Code 4

```
# More Libraries
import librosa.display
# Whole Process
mfc=librosa.feature.mfcc(y, sr=sr, n_mfcc=12)
plt.figure(figsize=(14,5))
librosa.display.specshow(mfc, sr=sr, x_axis='time', y_axis='hz')
# delta
d=librosa.feature.delta(mfc, width=9, order=1, axis=-1, trim=True)
plt.figure(figsize=(14,5))
librosa.display.specshow(d, sr=sr, x_axis='time', y_axis='hz')
# delta-delta
dd=librosa.feature.delta(mfc, width=9, order=2, axis=-1, trim=True)
plt.figure(figsize=(14,5))
librosa.display.specshow(dd, sr=sr, x_axis='time', y_axis='hz')
# Results
```

3.3 Deep Learning-Based Feature Extraction Techniques

Analysis done indicates that the use of deep learning for feature extraction emerged from 2011 [27]. It brings new era in the task of speech feature extraction. Usually, deep learning is applied in computer vision that involves images as dataset; however, CNNs indicate that if an audio file is presented appropriately, it is suitable for audio data also [28]. In case of deep learning-based feature extraction process, hidden layers of used deep learning model are extracted as the feature vectors. d-Vectors, j-vectors, x-vectors, etc., are categorized under deep learning-based speech feature extraction techniques [29]. Here is a brief discussion of some of these techniques.

3.3.1 d-Vectors

A lot of research has been done for extraction of deep layer's features. Variani et al. [30] trained a DNN with the perceptual linear predictive (PLP) features along with its delta and delta-delta vectors chosen for frame level. Designed DNN contains multiple fully connected layers. Desired d-vectors are the averaged output of activation function of these hidden layers. These features were used for the development of speaker recognition system.

3.3.2 x-Vectors

Hidden layers are extracted as feature vectors in the case of x-vectors. Deep learning network with time delay factor is preferred for extraction of these features. Fang et al. [31] designed a time delay neural network (TDNN) with multiple fully connected layers that operates on the frames of speech signal with some temporal context.

3.3.3 End-to-End Speech Signal

Generally, feature vectors are extracted from the speech signals, and these vectors are passed to the deep learning models for classification. However, deep learning classification models themselves should be trained to take a raw speech signal as input and to adapt it for further processing. These approaches are being applied in practice also these days [1, 32, 33].

4 Conclusion

Speech-based systems are current potential for the interaction with various intelligent devices. Speech recognition and speaker recognition are the two techniques that are applied for the interaction in various forms. This chapter discusses various sources of data available to assist the development of these systems, which revealed that publically available sources of data are most widely used and really helpful to develop the robust systems. Classical feature extraction techniques involve various speech signal processing steps to simulate the human auditory, shape of vocal tract, etc. Implementation details of these speech signal processing steps inspire for designing the new feature extraction techniques and for integration of features. Analysis of front ends of speech and speaker recognitions under clean and noisy environment indicates that integrated feature extraction techniques are improving the performance of these techniques. However, MFCC is the most popular feature extraction technique that is involved in integrated features as well as used individually. This chapter discusses MFCC with its implementation for static and dynamic coefficients. Deep learning is introducing a new era in the development of front end of speech-based features. This chapter discusses various deep learning-based feature extraction techniques that are potentially improving the performance of these systems.

References

1. M. Dua, R.K. Aggarwal, V. Kadyan, S. Dua, *Punjabi Speech to Text System for Connected Words* (IET, Bangalore, India, 2012)
2. M. Dua, R.K. Aggarwal, M. Biswas, GFCC based discriminatively trained noise robust continuous ASR system for Hindi language. J. Ambient. Intell. Humaniz. Comput. **10**(6), 2301–2314 (2019)
3. M. Dua, R.K. Aggarwal, M. Biswas, Discriminative Training Using Heterogeneous Feature Vector for Hindi Automatic Speech Recognition System, in *2017 International Conference on Computer and Applications (ICCA)*, (IEEE, Doha, 2017), pp. 158–162
4. T. Kinnunen, H. Li, An overview of text-independent speaker recognition: From features to supervectors. Speech Comm. **52**(1), 12–40 (2010)
5. J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, et al., State-of-the-art speaker recognition with neural network embedding's in NIST SRE18 and speakers in the wild evaluations. Comput. Speech Lang. **60**, 101026 (2020)
6. K. Kumar, H. Khalil, Y. Gong, Z. Al-Bawab, C. Liu, *U.S. Patent No. 10,706,852* (U.S. Patent and Trademark Office, Washington, DC, 2020)
7. M. Dua, R.K. Aggarwal, V. Kadyan, S. Dua, Punjabi automatic speech recognition using HTK. Int. J. Comput. Sci. Iss. (IJCSI) **9**(4), 359 (2012)
8. M. Dua, R.K. Aggarwal, M. Biswas, Discriminative training using noise robust integrated features and refined HMM modeling. J. Intell. Syst. **29**(1), 327–344 (2018)
9. M. Dua, R.K. Aggarwal, M. Biswas, Performance evaluation of Hindi speech recognition system using optimized filterbanks. Eng. Sci. Technol. Int. J. **21**(3), 389–398 (2018)

10. R.K. Aggarwal, A. Kumar, Discriminatively trained continuous Hindi speech recognition using integrated acoustic features and recurrent neural network language modeling. J. Intell. Syst. **30** (1), 165–179 (2020)
11. A. Kumar, R.K. Aggarwal, Hindi speech recognition using time delay neural network acoustic modeling with i-vector adaptation. Int. J. Speech Technol., 1–12 (2020). https://doi.org/10.1007/s10772-020-09757-0
12. M. Dua, R. Yadav, D. Mamgai, S. Brodiya, An improved RNN-LSTM based novel approach for sheet music generation. Proc. Comput. Sci. **171**, 465–474 (2020)
13. Q. Zhu, A. Alwan, Non-linear feature extraction for robust speech recognition in stationary and nonstationary noise. Comput. Speech Lang. **17**(4), 381–402 (2003)
14. V.Z. Këpuska, H.A. Elharati, Robust speech recognition system using conventional and hybrid features of MFCC, LPCC, PLP, RASTA-PLP and hidden Markov model classifier in noisy conditions. J. Comput. Comm. **3**(06), 1 (2015)
15. S. Ding, T. Chen, X. Gong, W. Zha, Z. Wang, AutoSpeech: Neural architecture search for speaker recognition. arXiv preprint **arXiv**, 2005.03215 (2020)
16. A. Lozano-Diez, A. Silnova, P. Matejka, O. Glembek, O. Plchot, J. Pesan, L. Burget, J. Gonzalez-Rodriguez, *Analysis and Optimization of Bottleneck Features for Speaker Recognition*, vol 2016 (Odyssey, Bilbao, 2016), pp. 352–357
17. P.S. Nidadavolu, J. Villalba, N. Dehak, Cycle-GANs for Domain Adaptation of Acoustic Features for Speaker Recognition, in *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, Brighton, 2019), pp. 6206–6210
18. V. Tiwari, MFCC and its applications in speaker recognition. Int. J. Emerg. Technol. **1**(1), 19–22 (2010)
19. U. Sarkar, S. Pal, S. Nag, C. Bhattacharya, S. Sanyal, A. Banerjee, D. Ghosh, Speaker recognition in bengali language from nonlinear features. arXiv preprint **arXiv**, 2004.07820 (2020)
20. A. Bouziane, J. Kharroubi, A. Zarghili, Towards an objective comparison of feature extraction techniques for automatic speaker recognition systems. Bull. Electr. Eng. Inform. **10**(1), 374–382 (2020)
21. M.N. Frankle, R.P. Ramachandran, Robust Speaker Identification Under Noisy Conditions Using Feature Compensation and Signal to Noise Ratio Estimation, in *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, (IEEE, Abu Dhabi, 2016), pp. 1–4
22. K.P. Bharath, R. Kumar, ELM speaker identification for limited dataset using multitaper based MFCC and PNCC features with fusion score. Multimed. Tools Appl. **79**(39), 28859–28883 (2020)
23. P. Alku, R. Saeidi, The linear predictive modeling of speech from higher-lag autocorrelation coefficients applied to noise-robust speaker recognition. IEEE/ACM Trans. Audio, Speech, Language Process **25**(8), 1606–1617 (2017)
24. P. Prithvi, T.K. Kumar, Comparative analysis of MFCC, LFCC, RASTA-PLP. Int. J. Sci. Eng. Res. **4**(5), 1–4 (2016)
25. M. Todisco, H. Delgado, K. Lee, M. Sahidullah, N. Evans, T. Kinnunen, J. Yamagishi, *Integrated Presentation Attack Detection and Automatic Speaker Verification: Common Features and Gaussian Back-End Fusion* (Interspeech, Hyderabad, 2018)
26. W. Cai, H. Wu, D. Cai, M. Li, The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion. arXiv preprint **arXiv**, 1907.02663 (2019)
27. N. Chen, Y. Qian, K. Yu, Multi-Task Learning for Text-Dependent Speaker Verification, in *Sixteenth Annual Conference of the International Speech Communication Association*, (Interspeech, Hyderabad, 2015)
28. S. Shuvaev, H. Giaffar, A.A. Koulakov, Representations of sound in deep learning of audio features from music. arXiv preprint **arXiv**, 1712.02898 (2017)

29. D. Sztahó, G. Szaszák, A. Beke, Deep learning methods in speaker recognition: a review. arXiv preprint **arXiv**, 1911.06615 (2019)
30. E. Variani, X. Lei, E. McDermott, I.L. Moreno, J. Gonzalez Dominguez, Deep Neural Networks for Small Footprint Text-dependent Speaker Verification, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, 2014), pp. 4052–4056. [d-vectors]
31. F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, J. Bonastre, Speaker anonymization using X-vector and neural waveform models. arXiv preprint **arXiv**, 1905.13561 (2019)
32. G. Heigold, I. Moreno, S. Bengio, N. Shazeer, End-to-End Text-dependent Speaker Verification, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, Shanghai, 2016), pp. 5115–5119
33. Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, L. Dai, Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system. Proc. Interspeech **2019**, 361–365 (2019)

Automatic Speech Recognition in English Language: A Review

Amritpreet Kaur, Rohit Sachdeva, and Amitoj Singh

1 Introduction

Humans can interact with one other or exchange information by speaking, writing, signing, or using some other ways. In this era of the computer, a natural language interface to computer plays a very important function for improving the usage of the computer for the common man. It is the demand of time to bring human-human interaction close to human-computer interaction as much as we can. Speech processing is one of the methods to provide the interaction between human and machine. Speech recognition, more popularly known as automatic speech recognition (ASR), has gained popularity since the last few years. ASR is a domain of machine learning that aids human beings to interact more easily and closely to the machines. ASR usefulness has become a fundamental piece of each application whether it is your cell phone to the grounded web application. The voice search feature is gaining great popularity nowadays. Voice recognition has been included in every app as an important integral functionality.

A. Kaur
Punjabi Universty, Patiala, Punjab, India

R. Sachdeva
M.M.Modi College, Patiala, Punjab, India

A. Singh (✉)
Jagat Guru Nanak Dev Punjab State Open University, Patiala, Punjab, India
e-mail: amitojsingh@mrsptu.ac.in

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_8

1.1 Speech Recognition Process

Speech comprises primarily three tasks: first, the comprehension of the vocabulary, which helps the device to remember terms, meanings, and sentences we express; second, the knowledge of the natural language, which leads the system to comprehend what we say; and, third, the symptoms of speech, which ensure the machine to recognize. The automated recognition of speech converts the captured signal into some sort of readable phrases, acts, or phonemes. The motive for speech acknowledgment is to build robots, which can be addressed when information is obtained. Speech recognition-based devices may provide speech to different posts, from learning aids, home technology, computer controls, and speech comprehension equipment to support individuals with physical disabilities.

New scientific experiments proved that speech is one of the simplest ways for man to communicate with devices [1]. However, we cannot provide language-based software such as mobile phones and transcription writing in a variety of fields. ASR uses a traditional model identification technique that maintains a sequence of characteristics to be learned in classes and compares network parameters with test models to identify them in the best matching design class. At early stage, ASR application study primarily involves the statistical models during speech recognition and has produced excellent results in multiple language forms across the globe. A scientifically developed ASR framework uses voice extraction approaches and uses voice parameters in audible models to add tentative details about the framework of the target language and to add voice extracted features and supposed theory phonetics. Language and pronunciation models are used to represent the language vocabulary and output the uttered test speech signal in text format using decoding techniques [2]. Notable development is also underway and creates a zero-loss speech recognition system using advanced technologies of feature elimination and sound modeling, large corpus of speech, different methods of language modeling, and better ways of decoding. [3–8].

HMM is a double deterministic process that is obtained by two interacting processes, a Markov chain with a limited number of states and a special statistical function that is paired with each of them in order to determine the possibility of acoustic characteristics. State variables can be modelled through distinct projections [9], semi-continuous ranges of probabilities [10], or continual allocations of probabilities [11]. Mixture distributions consisting of a linear interface of Gaussian or Laplacian probability density function (pdf) are widely used for discrete statistical simulations. Artificial neural network (ANN) is a modern architecture that is motivated by the organization of cells in the human brain. The multi-layer perceptron (MLP) is the preferred ANN model, which is used in speech recognition systems [12–14]. ANN with particular application of MLP helped the question classification in this platform. Some artificial intelligence methods have also been suggested in order to boost the state-of-the-art ASR programs [15].

As an advancement of ANN, deep learning methods are proposed to get significant improvement in the performance of the acoustic models. Restricted Boltzmann

machine, deep belief network, deep neural networks, convolutional neural networks, and capsule neural network are popular variants of deep learning technique successfully adopted for speech recognition tasks. They raise the performance of the system up to human level, but the main limitation is that they require large datasets for training and testing purpose.

1.2 Speech Recognition Architecture

The architecture of the ASR system that has two main components is broadly divided into five components as shown in Fig. 1.

1.3 Preprocessing and Feature Extraction

The role of speech recognition is to first convert the recorded analog speech signal to digital values by using analog-to-digital (A/D) converter. The A/D converter performs sampling, quantization, and encoding to convert an analog speech signal to digital values. A sampling rate of 16 kHz is used as a set standard. The amplitude of the recorded speech signal is taken at regular intervals of time to perform the sampling, and each sampled data output is quantized or approximated to a predefined value. This predefined value is further encoded into its equivalent digital form. Background noise removal, pre-emphasis, framing, and windowing are the four steps performed in the preprocessing phase of the ASR. Pre-emphasis filters the

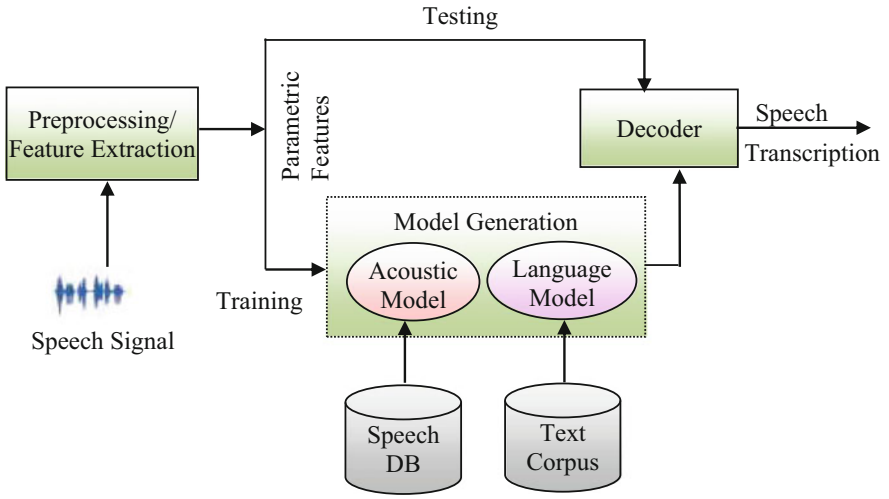

Fig. 1 The architecture of automatic speech recognition system

speech signal using a first-order finite impulse response (FIR). Framing divides the pre-emphasized speech signals into overlapped frames, where each frame is of size 10–25 ms and has 50–70% overlapping with its previous and successive frame to prevent loss of information. The process of windowing groups a number of such frames into one window. A stationary speech signal results in a more reliable spectral estimation than a variable speech signal. After pre-emphasis, the process of feature extraction or parameterization is applied to extract useful information from the speech signal [16, 17]). The function of this module is to find a set of acoustic correlated parameters from the recorded speech signal. Such parameters are termed features. These features are grouped together to form feature vectors. One of the objectives of parameterization is to acquire such information from the recorded speech signal that differentiates between the given sub-units of a text sequence [18]. Temporal analysis and spectral analysis are the two methodologies used for parameterization, where in the former speech signals are processed directly and in the latter features are computed by using short Fourier transformation [19].

The main objective of the feature extraction is to achieve a projection of the speech signal to a compact parameter space wherever the data associated with the speech content will be extracted simply.

1.4 Merit of Automatic Speech Recognition Systems

- They produce a large amount of writing in a comparatively short time.
- Reduced typing efforts.
- Fair accuracy.
- Dictation reduces the workload.
- Speech recognition technology is much faster than typing.
- ASR is very useful in a few areas like the medical sector.
- They help people with cognitive disabilities.
- Easily accessible datasets.
- Application is much secure in comparison to other security methods.
- Cost to operation is much low.

1.5 Demerits of Automatic Speech Recognition Systems

- Large amount of memory is required to store voice files.
- Due to noise interference, it is difficult to use in normal classroom settings.
- They require training of the software to recognize the voice for each user.
- Hard for poor decoders.
- Without adequate support, it can be frustrating for making errors.
- Automatic Speech Recognition Systems only assist at one stage of the writing process but do not provide the solution of the writing problem.

1.6 Applications of Automatic Speech Recognition Systems

Speech recognition is gaining popularity nowadays. Some of the commercial applications of ASR are as follows:

- Automated identification.
- Generating subtitles during the live telecast.
- Dictation in various professional areas.
- They provide medical assistance.
- Industrial robotics.
- Forensic and law enforcement.
- Defense and aviation telecommunications industry.
- Home automation and security access control.
- IT and consumer electronics.
- Speech to text conversion.
- Voice translation into foreign languages.
- Querying the database with spoken queries, e.g., E-Farming.
- Robotics.
- Facility of speech recognition in mobile applications.
- Different applications in embedded systems.
- Education and learning.

1.7 Different Approaches in Automatic Speech Recognition

There are various approaches that can be used for speech recognition. The following are few methods that can be used in the speech recognition process:

- *Pattern Recognition Approach*

Speech trends are described by a speech prototype or mathematical model (e.g., hidden Markov model or HMM). These speech trends can take several types.

- *Acoustic-Phonetic Approach*

According to acoustic-phonetic framework, there are different linguistic functions in the language that is spoken. These functions are broadly defined by a collection of acoustic characteristics that are modified over time. There are three approaches that have been taken to classify the language for solutions: Phone identification, Gaussian mixture model (GMM), and support vector machine (SVM), which helps in the classification of problems that are being investigated jn [6, 7].

- *Learning-Based Approach*

It also reduces the drawback of the HMM-based neural network and genetic algorithm training and design methods. This method utilizes intelligent learning through which it will learn by limitation.

- *Knowledge-Based Approach*

We assume that there would be substantial advances in explicit modeling. The knowledge-based method takes into account linguistic detail, phonetics, and spectrogram.

Artificial intelligence is a combination of the voice and pattern detection. Information would help the robots to solve problems better. Knowledge-based framework has been documented to improve innovation and produce improved solutions and designs.

- *Stochastic Approach*

Speech is totally unpredictable. Stochastic modeling approach enables the use of probabilistic models to deal with uncertain or incomplete information.

In this article, the authors reviewed the most relevant work in the literature including training techniques for recognizing speech models and especially concerning discriminative training of the DNN-HMM hybrid models for speech recognition and feature the most significant literature, which is essential in the discriminative learning of the DNN-HMM systems and feature extraction methodologies.

2 Challenges and Issues

The very broad study has been carried out of the existing literature to identify the various techniques and methodologies along with the existing challenges faced during speech recognition that could give rise to further research in the field. Speech recognition system can be separated in different classes with the mode of utterance that can be recognized. The speech can be divided into five categories. Front-end processes single utterance at a time. The connected word systems are identical to independent words but enable the utterances to be combined together with a limited delay. The program is difficult to get up and running because of special approaches used in understanding continuous expression. Spontaneous expression is very difficult to recognize compared to recorded expressions. It is expression that is normal, not memorized. An ASR system should be capable of managing mispronunciation, and other kinds of incomprehensible expression should be feasible. Still, the spontaneity in automatic speech recognition is a challenge. Due to this limitation of speech recognition, it gives birth to various challenges that may affect the performance of an ASR system. The challenges can be underlined like:

1. The background noise can be the problem for recognizing speech accurately in different environments and real-time applications.
2. The variation of the speakers in uttering a word can lead to pronunciation difference. This variation is further categorized into age, gender, and dialect of the physical appearance of the speaker.
3. A continuous speech system is rather difficult to implement when compared with the uninterrupted speech. We use continuous speech system in our real-life speech; this further poses problems in speech recognition systems.
4. Poor microphone quality, position, and direction of the microphone relative to the speaker.
5. Other factor contributing to the lower recognition rates includes pronunciation. This can also be a major challenge. It can be affected by the physiological aspect of pronouncing the words by the way in which stress is given on different syllables, phones, and vowels. This particularly affects the speech recognition of tonal languages.

Despite these challenges, environment variation, channel variation, style of speaking, age, gender, etc., are also the challenging task of speech recognition.

3 Motivation

To make an ASR system pervasive, it is crucial that the system should be capable of handling all types of variabilities such as language variabilities, speaker variabilities, and environment variabilities. Many state-of-the-art speech recognition systems and benchmark techniques have been proposed during the last six decades to make ASR systems independent to such variabilities. Despite all studies, the knowledge of the acoustic phonetic features of speech has not met the target, and 100% effective speech recognition system has not been realized. There are some strategies such as acoustical model adaptation and optimization of noise-robust features that can effectively incorporate noise-robust ASR systems.

Another significant argument is that ASR study works based on languages like English, Arabic, and Chinese. This is because most of the linguistics development center was based in these languages-speaking countries.

Most of the speech recognition algorithms use hidden Markov models (HMMs) and Gaussian mixture models (GMM) to evaluate speech variability. The scale of neural network is very large, and its use is very reliable. Deep neural networks have been able to outscore other classifiers, frequently used by research teams.

So many problems have been identified with the use of speech recognition systems. The training of deep neural networks (DNNs) with random initializations provides excellent results on the benchmark of ASR [20–22].

4 Inclusion-Exclusion Criteria

A systematic review of literature is a method to identify, evaluate, and interpret the available literature in the form of research papers, articles, journals, etc., so that the studied literature can be summarized, research gaps can be identified, and a base for carrying out future research can be formulated [23].

The datasets included in the studies are as follows:

- TIMIT
- CTIMIT
- NTIMIT
- SWITCHBOARD Data
- NIST (National Institute of Standards and Technology)
- DARPA
- CHiME
- OGI AlphaDigit
- IWAMSR
- Voice of America (VOA)
- FIRE
- IIIT-H
- NIST LRE (Language Recognition Evaluation) 2005
- NIST LRE 2006
- NIST LRE 2007
- KEELE Database
- IEMOCAP (Interactive Emotional Dyadic Motion Capture)
- DIRHA

From a large amount of existing literature related to speech recognition and its terminologies, the central idea was filtered with the help of database search using significant keywords. Primary research articles focused on the keywords included in different research papers. The selected papers were then refined manually by the authors. Irrelevant papers for the research were discarded manually based on the information in the abstract of the paper. The following facets were identified depending on the focus of the research and the research questions, which triggered the inclusion-exclusion principle for the research: speech recognition, feature extraction, large vocabulary continuous speech recognition, and ASR systems. The study followed a systematic approach to include quantitative and qualitative research articles, which were published till 2021. This made the database search more comprehensive. Finally, the primary studies were included in the review study depending on the abstract and the full text (Table 1).

Our randomized prospective method removed prejudices about the variability of design, diversity between study designs and community, and prejudice attributable to publication bias.

Table 1 Features for inclusion-exclusion criteria

Feature	Description
Speech recognition	The system is developed for speech recognition; it uses different approaches for speech recognition
Feature extraction	Its studies include different types of feature extraction techniques or hybrid feature extraction methods
Large vocabulary continuous speech recognition (LVCSR)	Emphasizes on speech recognition for LVCSR systems that include large vocabulary
ASR systems	Included results about the ASR systems so developed

5 Recognition of ASR in the English Language

The algorithmic methods that evaluate spoken utterances are based on the principle of method detection [24]. To reflect all limitations in a single modular deterministic finite-state network (FSN), top-down methods are taken, which are composed of audible HMM states with their outputs created by GMM phones and their coordinating arcs [11]. It is an a posteriori technique to find the best chain of word from the FSN as the known expression for a defined input speech.

The search strategy method has obtained successful outcomes in many search activities (Table 2).

Most of the latest voice recognition technologies use hidden Markov models (HMMs) to evaluate how well a sample or a short range of utterance that is acoustic inputs suits to each state of each HMM. A way to test the statistical fit is to use a feedforward neural network (FNN) that takes a group of samples of utterances as input and generates residual chances for each HMM state as output. Deep neural networks have been able to outscore other classifiers, frequently by wide margins.

6 English Language

English is an American-British language that has been accepted globally and spoken by people around the world for communication. The English language has mainly two types, namely, British English and American English. Machine learning may allow major changes in automated speech recognition. The greatest advance took place nearly 40 years ago, when the expectation maximization (EM) algorithm was discovered for training hidden Markov models (HMMs). The EM algorithm allows real-world speech recognition possible by considering the associations between speech states and audible inputs as Gaussian mixtures. In these structures, the audible input is provided by the combination of Mel Frequency Cepstral Coefficients (MFCCs) and perceptual linear prediction (PLP) calculated from the waveform [25, 26]. An effort has been made to convey the remaining details to promote prejudice in type of GMM-HMMs. Imseng et al. [27] tested the ASR for multilingual speech. They used Corpus Speech Data (II) only, which requires three single words

Table 2 Analysis of feature extraction techniques

Feature extraction	Merits	Demerits
DWT	Includes only the temporal information along with the frequency information which is adequate for the localization of time and frequency that used in noisy environments	Inflexible owing to the use of similar basic wavelets for the speech samples
MFCC	Coefficients are loosely related Linear characteristics are not considered Reliable discrimination efficiency	Affected by noise Phase spectrum is not considered
WPT	High-frequency samples are taken into account	Inflexible owing to the similar basic wavelets for the speech samples
LPC	Low-dimensional feature vectors used to represent spectral envelope Computationally correct and easily implementable Static feature extraction	Unsuitable linear scaling for speech perceiving speech signals No prior knowledge of the speech sample included
LPCC	Decorrelation among feature vectors More robust than LPC	Unsuitable linear scaling for speech perceiving speech signals No prior knowledge of the speech sample included
PLP	Reduced dissimilarity among voiced and unvoiced speech Does not depend on vocal tract length	Affected by noise, equipment, and the channel
RASTA-PLP	Spectral factors are reduced Robust technique	Performs poor in noise-free conditions
VQ	Takes up lesser storage for storing spectral analysis information Trains at a faster rate Speech signals are modeled in a discrete manner	Training time is directly proportional to the vocabulary size Temporal information not considered Involves computation errors in quantization

per speaker for each application. The framework is used to define a category of approximately 30 words for use in immersive voice response systems, such as "help" and "cancel." The authors have developed knowledge phoneme-based discrete speech perception systems, which design a three-state left-to-right HMM for each knowledge phoneme. For the training and identification of GMM systems, an HTK toolkit [28] is used, in which 32 mixtures of Gaussian with diagonal covariance matrices are modelled for each condition. They studied the efficiency of speech recognition systems, which have multiple functions and audible processing methods in a mixed challenge in languages (where the meaning of the dialect of the test speech is expected to be unidentified). Soltau et al. [29] demonstrated a basic move in machine learning to expand the maximal utilized design of a linear layer to an undefined framework of the graph. This move incorporates the advantages of convolution neural networks with the advantages of normal networks. The joint model has only a remarkable improvement in parameters and is virtually unchanged in preparation and decoding time. In two LVCSR tasks and a speech recognition task

the authors have reported substantial changes over the parameters. Limited changes (18.9% to 18.6% WER), though almost doubling the parameter scale, were recorded. This model has a different architecture: it is designed to share the majority of layers across MLP and CNN. Therefore, the model provides the same input capability for both CNN and MLP [30]. Seide et al. [30] applied context-dependent deep-neural-network HMMs to speech-to-text transcriptions. The word error rate achieved by GMM is decreased from 27.4% to 18.5%, which is a relative gain of 33% on switchboard (RT03S) corpus. CD-DNN-HMMs are stacked with conventional tied-state triphones and deep-belief pretraining networks. In the past, word error rate (WER) was minimized by 16% tied states in tens of hours of preparation. This has contributed to the further implementation of CD-DNN-HMMs and reveals how sparing can be used for the transcription utilizing over 300 h of training data, 9000 tied-up states having up to nine hidden layers. Relative error reductions of 22.28% are observed for four less compatible transcription activities. Apart from conventional speech recognition models, deep learning methods have been successfully applied by Hinton et al. [20]. DNN speech recognition functions have been qualified in phoneme recognition tasks [31]. They observed that phonetic characteristics such as articulation, position of articulation, and voicing narrowly affect the performance. Nowadays, because of their substantial enhanced efficiency over other models, DNN is the most popular technology for acoustical phonetic modeling. However, the computation they used to construct phonemic groups from highly variable acoustic signals is not very well developed. They observed a DNN to identify the performance phoneme and defined both at the single and community levels on each layer with recognition characteristics. Node related to particular articulation methods and position was removed from the first secret layer, and deeper layers became more sparse. Different phonetic characteristics in all layers is observed with high sparse at the each level of the node. In addition, they observed that nodes with identical phonetic characteristics were triggered differentially into different instances. This study shows that phonetic characteristics coordinate activations at various levels of the DNN, which represents the conclusions of the encoding of functions in the human auditory system. A simple modification of neural networks implies expanding to an arbitrary graph structure the widely used linear layer structure. Soltau et al. [32] stated that the basic modification of the neural network is the extension of a widely used linear layer system which is employed into a single graphics structure by the subjective graphics firm. In two LVCSR tasks and a voice action detection task, substantial changes are reported over very high baselines. The authors demonstrated that the graphic framework has the advantage of helping one to jointly train convolution and conventional neural networks. The joint training method does not enable I-vectors to take advantage of convolutional neural networks because I-vectors are not topographical features. The error rate has reduced from 11.8% to 10.4% by the baseline CNN. The analysis showed that a relative gain is 10% over the previous system. The authors have shown that their model fits for various tasks such as LVCSR and RATS keyword search. State-of-the-art speech recognition technologies now include deep neural networks. Design neural network-based acoustic models involve a range of design decisions including layout of the

network, size, and training. An empirical analysis was carried out on the aspects of DNN-based acoustic model architecture, which are most relevant for the high quality of speech recognition systems. DNN-based speech recognition systems have introduced tremendous enhancements to the LVCSR in recent years. The first study presumed that DNNs perform well due to unmonitored pretraining [33]. In LVCSR, many speech recognition benchmarks have been achieved by DNNs with random initialization [20–22]. A state-of-the-art method of speech recognition built by deep learning offers performance near to human beings. The design is considerably simpler than conventional systems that rely on states of the HMM. This conventional solutions are even less successful in noisy settings (A [34]). By concluding the details from 174 articles released from 2006 to 2018, Nassif et al. [35] submitted a comprehensive statistical study of the usage of deep learning in speech applications. The most prominent application of deep learning algorithms has been to improve computer skills to realize what people should do like speech recognition.

Hansen and Mirsamadi [36] found international language-accented language as an interpolation between native language (L1) and English (L2). The paradigm used is upgraded for accentual expression in a manner that could concurrently answer both languages. They investigated the usage of native-language data for better results in accentuated English language through an end-to-end recurrent neural network (RNN) framework trained in English and Indian language. In this relation, we analyze pretraining with native languages as well as multi-task learning (MTL). The findings indicate that the proposed MTL model works better than pretraining and outperforms a simple model that is trained simply with English data. In the same way, the authors examined pretraining with native languages. The authors introduced a modern MTL environment in which both English and a native language are educated in the secondary role and the same performance set is used. Their model showed English outcome with further focus on self-employed results and increased model quality to a relative character error rate (CER) of +11.95% and +17.55%. Audhkhasi et al. [37] said that direct acoustic-to-word (A2W) modeling demands more data orders to be able to compete competitively than subword unit models. They studied an A2W model on a 2000-h data collection, which is compatible with other sophisticated hybrid and end-to-end models utilizing subword units. The proposed SR model provides the user a rich and understandable performance while retaining the ease of learning and decoding of the A2W model (Tables 3 and 4).

7 Synthesis Analysis

From the literature studied so far, the following findings have been listed. The most relevant finding here is that to attain better accuracy, a researcher should use hybrid feature extraction techniques. Such techniques provide efficient and useful information for the input to a classifier.

Table 3 Some important work on the English language

Author	Type of Corpus	Feature extraction	Classification	LM	DB	Duration	WER %
[38]	Continuous speech recognition	CNN	CNN	–	TIMIT	4 h	20.36
[39]	Continuous speech recognition	–	Connectionist temporal classification (CTC) and RNN	Tri-gram	Wall Street Journal	81 h	6.7
[40]	Telephonic conversation	PLP and I-vector	Joint CNN/DNN	N-gram+NN LM	Switchboard	2000 h	8.0
[41]	Continuous speech recognition	Log Mel filter bank	BLSTM RNN, BLSTM CTC	N-gram	Switchboard	2000 h	8.5
[42]	Continuous speech recognition		CTC-RNN	N-gram	11,940 h (Eng)	Self-created	3.10 (clean) 21.59 (noisy)
[8]	Telephonic conversation	f-MLLR+I-vector	RNN + DeepCNN +BLSTM	N-gram+NN LM	Switchboard	2000 h	6.6

Table 4 English language literature review on deep learning

Ref	Corpus	Dataset	Feature extraction Technique	Language model	Acoustic modeling	WER (%)
[29]	DARPA RATS, SWB-l	256 frames	LVCSR,I-vector	N-gram	MLP,CNN, MLP/CNN	10.4
[43]	WSJ Aurora 4	5 hidden layers, 2048 nodes each, an output corresponding to 1209 senones, 3 and 12 phones	HMM, LVCSR, KWS	N-gram	DNN, NN-HMM	12
[44]	Switchboard and Fisher	8986 output classes	HMM-DNN, LVCSR, GMM	Spectrogram	HMM-GMM, DNN	20.7
[45]	TIMIT	39 phonemes, 32 utterances	NN-HMM	Bigrams, triphone	GMM-HMM, DNN-HMM, RNN	16.88
[34]	Switchboard Hub5'00	30,000-word vocabulary, 5 hidden layers each with 2048 neurons	NN-DNN	N-gram	RNN, HMM	16.0
[46]	Switchboard	37 K sentences	HMM	Bigram, extended tri-gram, trigram	LVCSR, HMM-DNN, RNN	18.6, 11.7, 10.8, 9.3
[47]	BlackOut	71,350 words and 3720 unique words	RNN	N-gram	RNN, DNN, LSTM	51.3
[48]	Numbers'95	27 phones, 3 hidden layers each having 1024 nodes	DNN	–	DNN-HMM	15.4
[49]	1B word benchmark	793,471 words	RNN	N-gram	RNN, LSTM	30.0
[50]	TIMIT	3696 utterances from 462 speakers, 24 speakers (2 males, 1 female)	TDNN	Bigram	LSTM	16.49
[51]	Switchboard	–	ANN, HMM	Trigram	CD-DNN-HMM, GMM	27.4
[50]	Aurora 4	7137 utterances from 83 speakers	HMM, GMM-HMM	Bigram	CD-DNN-HMM	22.5

(continued)

Table 4 (continued)

Ref	Corpus	Dataset	Feature extraction Technique	Language model	Acoustic modeling	WER (%)
[52]	CallHome, Switchboard, GigaWord	61 k word	HMM, GMM	Bigram	HMM-GMM, SGMM	54.7
[37]	Switchboard-Fisher	300 h 262 h, 2000 h 1698 h	DNN, HMM	N-gram	LSTM (BLSTM) RNN	8.8%/ 13.9

The method to combine different classifiers depends on the information which is provided by a single classifier.

Owing to the variations in the style of speaking of different individuals of different regions, speech recognition has definitely been a demanding research area. The much more work needs to be done on continuous speech.

Since the performance of a classifier relies on the extraction of the features at the feature extraction stage, it is thus important to carefully select the methods of feature extraction and the classifiers.

The lack of standard databases available for the researchers is significantly low. This presents a future direction to perform various experiments.

The accuracy of speech recognition and identification process relies on the features that have high discriminating power. Thus, there is a drastic requisite to study variant feature selection algorithms which can achieve good accuracy.

It is observed that the most commonly used feature extraction methods for speech recognition and identification task are PLP, LFCC, MFCC, and RASTA.

It is also noted that the most commonly used classifiers for speech recognition and identification task are HMM, GMM, DNN, and DNN-HMM.

The HMM classifiers are commonly used for speech recognition with good accuracy and this thing is also obsereved from the literature studied so far. The toolkits commonly used by the researchers are Sphinx and HTK. But again, some findings proved that accuracy rates can be increased with DNN, DNN-HMM, HMM-GMM, and ANN models.

8 Discussion and Future Scope

In the field of speech recognition, an immense number of directions can be explored to carry out further research. The proposed techniques being used for extracting the features of the speech samples can be extended further by combining different techniques to improve the speech recognition rate and WER.

The researchers should develop standard databases for various languages. The LVCSR database systems should be focused more. These databases should be made accessible to the small researchers so that future research can nurture.

Indian languages have not been employed in efficient feature extraction techniques. Researchers have only focused on baseline MFCC features as studied in the prevalent literature. Hybrid features like MFCC+LDA+MLLT, MFCC+BFCC +GFCC, LDA+MFCC, MF+PLP, RASTA-PLP, etc., may be applied on Indian languages.

More research can be carried out on the refinement techniques and recognition methodologies. A technique should be proposed that optimizes the acoustic feature selection. Researchers in the literature have reported their results of recognition accuracy in clean environments (i.e., noise-free). However, in real-time applications, the background noise and other factors affect the accuracy of recognizing the speech. Research on extracting efficient and suitable features and noise cancellation should be conducted for developing ASR systems.

9 Conclusion

In this chapter, the authors have presented a broad review and analysis of different feature extraction techniques employed for speech recognition for the English language. Different evaluation parameters have been summarized by the researchers from the wide research done by them. Further researchers reported their work onthe basis of the work done by them. The work done by researchers has been mentioned with the help of different tables. These parameters were WER, mWER, SER, PER, accuracy, recognition rate, and comparative analysis of different techniques. After reffering to the wide literautre the authors have suggested that the efficient feature extraction techniques needs to be implemented for the English language. This is because the research area of speech is tremendously wide. Also, not many accurate ASR systems have been developed for the English language, which can be efficient for continuous speech. Furthermore, LVCSR systems should be explored more by the researchers in order to improve the accuracy of recognizing speech for such systems so that applications for real-time use can be developed.

References

1. E. Yücesoy, V.V. Nabiyev, A new approach with score-level fusion for the classification of a speaker age and gender. Comput. Electr. Eng. **53**, 29–39 (2016)
2. R.K. Aggarwal, M. Dave, Performance evaluation of sequentially combined heterogeneous feature streams for Hindi speech recognition system. Telecommun. Syst. **52**(3), 1457–1466 (2013)

3. A. Adiga, M. Magimai, C.S. Seelamantula, Gammatone Wavelet Cepstral Coefficients for Robust Speech Recognition, in *Paper presented at the TENCON 2013–2013 IEEE Region 10 Conference (31194)*, (IEEE, Xi'an, 2013)
4. R.K. Aggarwal, M. Dave, Discriminative Techniques for Hindi Speech Recognition System, in *Information Systems for Indian Languages*, (Springer, 2011), pp. 261–266
5. V. Kadyan, S. Shanawazuddin, A. Singh, Developing children's speech recognition system for low resource Punjabi language. Appl. Acoust. **178**, 108002 (2021). https://doi.org/10.1016/j.apacoust.2021.108002
6. V. Passricha, R.K. Aggarwal, Convolutional support vector machines for speech recognition. Int. J. Speech Technol. **22**(3), 601–609 (2019a)
7. V. Passricha, R.K. Aggarwal, End-to-end acoustic modeling using convolutional neural networks. Intell. Speech Signal Process., 5–37 (2019b)
8. W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, A. Stolcke, The Microsoft 2017 Conversational Speech Recognition System, in *Paper Presented at the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, Calgary, AB, 2018)
9. M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, D. Van Compernolle, Template-based continuous speech recognition. IEEE Trans. Audio Speech Lang. Process. **15** (4), 1377–1390 (2007)
10. X.D. Huang, M.A. Jack, Semi-continuous Hidden Markov Models for Speech Signals, in *Readings in Speech Recognition*, (Elsevier, Amsterdam, 1990), pp. 340–346
11. S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Syst. Tech. J. **62**(4), 1035–1074 (1983)
12. A. Arora, V. Kadyan, A. Singh, Effect of Tonal Features on Various Dialectal Variations of Punjabi Language, in *Advances in Signal Processing and Communication: Select Proceedings of ICSC 2018*, ed. by B. S. Rawat, A. Trivedi, S. Manhas, V. Karwal, (Springer, New York, 2018), pp. 467–472
13. Kumar, Y.,Singh, N., Kumar, M., Singh, A., AutoSSR: An efficient approach for automatic spontaneous speech recognition model for the Punjabi language, Soft Comput., Springer, 25, 1617–1630 2020 https://doi.org/10.1007/s00500-020-05248-1
14. S. Masmoudi, M. Frikha, M. Chtourou, A.B. Hamida, Efficient MLP constructive training algorithm using a neuron recruiting approach for isolated word recognition system. Int. J. Speech Technol. **14**(1), 1–10 (2011)
15. P. Pujol, S. Pol, C. Nadeu, A. Hagen, H. Bourlard, Comparison and combination of features in a hybrid HMM/MLP and a HMM/GMM speech recognition system. IEEE Trans. Speech Audio Process. **13**(1), 14–22 (2005)
16. J. Kaur, A. Singh, V. Kadyan, Automatic speech recognition system for tonal languages: State-of-the-art survey. Arch. Comput. Method. Eng., Springer **28**, 1039–1068 (2020). https://doi.org/10.1007/s11831-020-09414-4
17. A. Singh, V. Kadyan, M. Kumar, N. Bassan, ASRoIL: A comprehensive survey for automatic speech recognition of Indian languages. Artif. Intell. Rev.., Springer **53**, 3673–3704 (2019)
18. J.W. Picone, Signal modeling techniques in speech recognition. Proc. IEEE **81**(9), 1215–1247 (1993)
19. W. Ghai, N. Singh, Phone based acoustic modeling for automatic speech recognition for punjabi language. J. Speech Sci. **1**(3), 69–83 (2013)
20. G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
21. B. Kingsbury, T.N. Sainath, H. Soltau, Scalable Minimum Bayes Risk Training of Deep Neural Network Acoustic Models Using Distributed Hessian-Free Optimization, in *Paper Presented at the Interspeech*, (Interspeech, Portland, OR, 2012)
22. K. Veselý, A. Ghoshal, L. Burget, D. Povey, Sequence-discriminative training of deep neural networks. Paper presented at the Interspeech. **arXiv**, 1808.00639v1 (2013)

23. B. Kitchenham, S. Charters, *Guidelines for Performing Systematic Literature Reviews in Software Engineering* (CiteSeerX, Princeton, NJ, 2007)
24. L. Lee, R.C. Rose, Speaker Normalization Using Efficient Frequency Warping Procedures, in *Paper Presented at the Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference*, (IEEE, Atlanta, GA, 1996)
25. S. Furui, Speaker-independent isolated word recognition using dynamic features of speech spectrum. IEEE Trans. Acoust. Speech Signal Process. **34**(1), 52–59 (1986)
26. H. Hermansky, Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. **87**(4), 1738–1752 (1990). https://doi.org/10.1121/1.399423
27. D. Imseng, H. Bourlard, M.M. Doss, Towards Mixed Language Speech Recognition Systems, in *Paper Presented at the Eleventh Annual Conference of the International Speech Communication Association*, (Idiap, Switzerland, 2010)
28. S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, et al., *The HTK Book*, vol 3 (Cambridge University Engineering Department, Cambridge, 2002), p. 175
29. H. Soltau, H.-K. Kuo, L. Mangu, G. Saon, T. Beran, Neural Network Acoustic Models for the DARPA RATS Program, in *Paper Presented at the INTERSPEECH*, (ISCA, Lyon, France, 2013)
30. F. Seide, G. Li, D. Yu, Conversational Speech Transcription Using Context-Dependent Deep Neural Networks, in *Paper Presented at the Twelfth Annual Conference of the International Speech Communication Association*, (ISCA, Florence, Italy, 2011)
31. T. Nagamine, M.L. Seltzer, N. Mesgarani, Exploring How Deep Neural Networks Form Phonemic Categories, in *Paper Presented at the Sixteenth Annual Conference of the International Speech Communication Association*, (ISCA, Dresden, Germany, 2015)
32. H. Soltau, H. Liao, H. Sak, Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. arXiv preprint **arXiv**, 1610.09975 (2016)
33. G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. **20**(1), 30–42 (2012)
34. A. Hannun, A. Maas, D. Jurafsky, A.Y. Ng, First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. arXiv preprint **arXiv**, 1408.2873 (2014)
35. A.B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech recognition using deep neural networks: A systematic review. IEEE access, **7**, 19143–19165 (2019)
36. S. Mirsamadi, J.H. Hansen, A Study on Deep Neural Network Acoustic Model Adaptation for Robust Far-Field Speech Recognition, in *Paper Presented at the Sixteenth Annual Conference of the International Speech Communication Association*, (ISCA, Dresden, Germany, 2015)
37. K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, M. Picheny, Building Competitive Direct Acoustics-to-Word Models for English Conversational Speech Recognition, in *Paper Presented at the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, Calgary, 2018)
38. O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(10), 1533–1545 (2014). https://doi.org/10.1109/taslp.2014.2339736
39. A. Graves, N. Jaitly, Towards end-to-end speech recognition with recurrent neural networks. Paper presented at the International Conference on Machine Learning. PMLR **32**(2), 1764–1772 (2014)
40. G. Saon, H.-K.J. Kuo, S. Rennie, M. Picheny, The IBM 2015 English conversational telephone speech recognition system. arXiv preprint **arXiv**, 1505.05899 (2015)
41. T. Yoshioka, M.J. Gales, Environmentally robust ASR front-end for deep neural network acoustic models. Comput. Speech Lang. **31**(1), 65–86 (2015)
42. D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, et al., Deep Speech 2: End-to-End Speech Recognition in English and Mandarin, in *Paper Presented at the International Conference on Machine Learning*, (arXiv.org, 2016)

43. X. Chen, X. Liu, M.J. Gales, P.C. Woodland, Recurrent Neural Network Language Model Training with Noise Contrastive Estimation for Speech Recognition, in *Paper Presented at the 2015 IEEE International Conference on Acoustics, Speech and, Signal Processing (ICASSP)*, (IEEE, Brisbane, 2015)
44. A.L. Maas, P. Qi, Z. Xie, A.Y. Hannun, C.T. Lengerich, D. Jurafsky, A.Y. Ng, Building DNN acoustic models for large vocabulary speech recognition. Comput. Speech Lang. **41**, 195–213 (2017)
45. J. Chorowski, D. Bahdanau, K. Cho, Y. Bengio, End-to-end continuous speech recognition using attention-based recurrent NN: First results. arXiv preprint **arXiv**, 1412.1602 (2014)
46. D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-End Attention-based Large Vocabulary Speech Recognition, in *Paper presented at the Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference*, (IEEE, Shanghai, 2016)
47. S. Ji, S. Vishwanathan, N. Satish, M.J. Anderson, P. Dubey, Blackout: Speeding up recurrent neural network language models with very large vocabularies. arXiv preprint **arXiv**, 06909 (2015)
48. P. Dighe, G. Luyet, A. Asaei, H. Bourlard, Exploiting low-Dimensional Structures to Enhance DNN based Acoustic Modeling in Speech Recognition, in *Paper Presented at the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, Shanghai, 2016)
49. R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, Y. Wu, Exploring the limits of language modeling. arXiv preprint **arXiv**, 02410 (2016)
50. J. Michalek, J. Vaněk, A Survey of Recent DNN Architectures on the TIMIT Phone Recognition Task, in *Paper Presented at the International Conference on Text, Speech, and Dialogue*, (arXiv.org, 2018)
51. F. Seide, G. Li, X. Chen, D. Yu, Feature Engineering in Context-dependent Deep Neural Networks for Conversational Speech Transcription, in *Paper Presented at the Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop*, (IEEE, Waikoloa, HI, 2011)
52. L. Burget, H. Heřmanský, Data Driven Design of Filter Bank for Speech Recognition, in *Paper presented at the International Conference on Text, Speech and Dialogue*, (Springer, Switzerland, 2001)

Noise-Robust Gender Classification System Through Optimal Selection of Acoustic Features

Puneet Bawa, Vaibhav Kumar, Virender Kadyan, and Amitoj Singh

1 Introduction

A systematic study of the sounds acoustic in speech sounds used for gender detection and their relationship to interpretation is a challenging objective that in many areas, from linguistically to computer recognition, have important implications and applications. Human speakers typically use a normal system in which air is discharged from the lungs and formed into vocal cords and organs, including the tongue, lips, teeth, etc. [1]. Likewise, the acoustic voice analysis depends upon the sample characteristic parameters such as filtering, power, frequency, and duration [2]. These acoustic features have been traditionally defined mainly through the implementation methodologies of linear analytical and visualization approaches. However, in recent years, it is clear that these spectral representations were only very crude approximations to those actually produced by the auditory path in the peripheral and central regions [3–5]. Some of the most important characteristics of the auditory images are due to the asymmetric form of the cochlear filters and the retention of the fine-temporal filter output structure below 3–4 kHz [6]. Likewise, two main reasons for applying reliable biophysical models of the auditory system are

P. Bawa · V. Kumar
Centre of Excellence for Speech and Multimodal Laboratory, Chitkara University Institute of Engineering & Technology, Chitkara University, Chandigarh, Punjab, India
e-mail: puneet.bawa@chitkara.edu.in

V. Kadyan
Speech and Language Research Centre (SLRC), School of Computer Science, University of Petroleum & Energy Studies (UPES), Dehradun, Uttarakhand, India
e-mail: vkadyan@ddn.upes.ac.in

A. Singh (✉)
Maharaja Ranjit Singh Punjab Technical University, Bathinda, Punjab, India
e-mail: amitojsingh@mrsptu.ac.in

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2_9

detailed in the concepts relating to auditory systems. First, the relation of the acoustic characteristics of the speech to conventional systems of phonetic classification [7, 8] (as expressed by their output patterns) should be described. Second is the need of defining the practical standards that permits correct encoding of the speech signal in the presence of high background noise and over a wide spectrum of sound pressures [9]. In the same way, a system can also be taught when to use the robust machine learning algorithms to select and incorporate the functionality necessary for mapping voice data.

With technology growing rapidly, machine learning is an area of science that has undergone significant changes and is also a common trend [10]. Machine learning is an artificial intelligence subcategory that uses algorithms and data for computer learning to decide on particular issues in different areas, such as accounting, finance, and medicine, to recognize gender, voicing by means of machine learning and data mining techniques. In addition, the role of gender recognition applies to various digital multimedia applications including speech recognition, intelligent human-computer interactions, speaker diarization, biometrics, and video indexing [11–13]. Likewise, considering the rising demand of machine learning applications in speech recognition topic, the methodology to efficiently recognize gender has played an important role in the field of healthcare systems with existence of some pathologies including vocal-fold cyst. However, it is still regarded as a very difficult and daunting challenge to build a predictive model for gender identification through speech [14]. Also, in such a quickly developing environment of computerization, one of the most vital problems in the developing world is correct gender identification in Indian native languages, which are often termed the low-resource languages [15–17]. However, it is also a costly and time-consuming challenge to find enough marked data for classified training classifiers to make precise predictions, so human work is required, although it is much easier to find unlabeled data in general. Semi-supervised learning (SSL) algorithms are considered to be an appropriate way of exploiting secret data in an unlabeled collection to develop more precise classifications in order to tackle the issue of inadequateness existing in the low-resource data [18, 19]. In a similar manner, many classes of SSL algorithms have been proposed with each being evaluated on different methodologies and approaches with an objective of finding adequate relational difference in the distribution of labeled and unlabeled data.

There are several approaches to speech synthesis that can be used to enhance the incoming speech signal. Similarly, the work to be performed has to result in ground realities to match real-time system implementations and applications. Taking such real-time situations into account, in this chapter, noise data augmentation technique has been applied to introduce into the original dataset using three distinct types of noises, including Babble, Factory, and Volvo at random SNR values and labeled as male and female for classification. Further, this chapter uses a warbleR library package [20] with an objective of performing the acoustic analysis for visualizing the process of gender detection in dialectal Punjabi language. As of our knowledge, some efforts had been made for the development of adequate language resources, but no effort has been made in designing the classifiers corresponding to the Punjabi

children speech. Moreover, the study to access adequate dataset has been performed with the findings comparable with gender detection in order to optimize the selection of required parameters among the extracted 20 acoustic parameters. Finally, the adequate model for recognizing the gender based on the optimal selection of the extracted acoustic features has been made through the comparative analysis of three machine learning algorithms including random forest, SVM, and MLP.

2 Related Work

Analyzing audio and extracting features sometimes can become a significant task when you have to pick certain features and reject in order to perform some tasks. In [21], the authors used machine learning algorithm and computed features that can help to check the authenticity of the audio signal. The experiments were able to distinguish appropriate value for hyper-parameters to be used. Li and Liu [22] experimented with Mel filter energy features and Mel Frequency Cepstral Coefficient (MFCC) features as acoustic criterion for detecting Mandarin vowels with low error rate and high investigation rate. In [23], authors also explored selecting optimal features for accent recognition using MFCC, spectrogram, and spectral centroid features extracted from audio samples and fed the features into two-layer convolutional network. The results depicted that MFCC feature yields the highest accuracy. Likewise, authors in [24] also explored predicting the reason for a newborn baby based on acoustic features. Pitch features and formant frequencies chosen as acoustic features alongside K-means algorithm proved quite handy and provided conclusive results for detecting a "pain" in cry along with reason for the cry. Likewise, the research endeavors for building the state-of-the-art speech recognition model in tonal languages have been analyzed on the basis of the findings relating to native languages [11]. In [25], authors proposed an automated attendance system using audio for gender classification and image for matching the current visual with the stored one in database in order to evaluate whether a student is actually present in the class or not. In [26], investigators explored gender modeling with clean and noisy environments and presented MFCC features alongside Gaussian mixture model (GMM) for audio modeling. The proposed system was capable of gender classification based on either audio or visual feedback whichever is less noisy, although the method is vulnerable to a scenario when both audio quality and visual quality are bad; that is, data is noisy. For simulating male/female detection, in [27], authors investigated GMM modeling along with pitch parameters and RASTA-PLP variables. Both clean and noisy environments were considered while evaluating the generated GMM, which was obtained by varying covariance matrix. The proposed method seems as a step in right direction. Likewise, Copiaco et al. [28] also experimented with multi-channel audio classification with MFCC and Power Normalized Cepstral Coefficients using deep convolutional neural network. The proposed methodology produced 98% accuracy. In [29], authors stacked different machine learning models and tried to use acoustic features to model the data. A

slight improvement in accuracy has been observed with state-of-the-art methods, but it came with a space complexity for such a stacked model alongside time complexity for predicting the gender on one sample. In [30], authors experimented with Mel Frequency Spectral Coefficients (MFSC) rather than MFCC features and used simple neural network for classifying the data based on gender. The selection of optimal features and parameters proved decisive at the end as the results showed substantial improvement in accuracy with smoothing applied.

Using deep learning algorithms dynamically selects essential information in raw language signal for processing of classification layer. Thus, with the proposed algorithm [31], the researcher has avoided the absence of knowledge on feeling, which cannot be modeled mathematically as more of an acoustic feature of voice. In [32], research was conducted in Bahasa Indonesia related to gender identification, and a supervised machine learning algorithm was applied with MFCC features with several modeling algorithms such as SVM, K-nearest algorithm, and artificial neural network. The results paved the way for impact analysis of gender identification for audio recognition. In [33], authors experimented with long short-term memory (LSTM)-based recurrent neural networks for predicting age and gender using audio sample and also reduced the over-fitting problem by using data augmentation and improved the testing accuracy using regularization. The authors also explored bidirectional LSTM alongside MFCC features on low resource dataset and found that more data can yield more accurate results [34]. Also, assembling modeling techniques have been explored using machine learning models like naive Bayes, random forest, and linear regression for hate speech detection by processing Twitter dataset. The study shows that such kind of models can help achieve adequate results [35]. Analysis of audio features was also performed by researchers, and they found that algorithms such as gradient boosting and random forest can help in classifying gender based on acoustic features [36]. The researchers also set up a pipeline for gender-based emotion recognition where MFCC features along with convolutional neural networks were used with an average pooling layer instead of a fully connected layer at the end can achieve accurate results [37].

3 Semi-supervised Classification Algorithms

3.1 Random Forest

Random forest classifier is known for its best use in classification and regression tasks. It is an ensemble algorithm that utilized a stack of decision trees and predicted the class or probability value for every node in the tree. It is often known as random decision tree. On the other hand, the trees can be allotted a certain weight depending upon the importance of node in the decision tree. The node yielding low error rates has the chance of accurate predictions and hence should be allotted higher weight and vice versa. Setting up such pipelines can end up outputting decisive predictions.

3.2 Support Vector Machine

Support vector machine is a supervised modeling approach, which is known as one of the best in classifying or regression problem analysis. SVM models the training samples in such a way that it maximizes the difference between two given classes. A new sample is mapped to a space, and then the modeling algorithm tries to predict whether the sample belongs to the allotted space or not.

For a given training sample

$$(a_1, b_1), (a_2, b_2), \ldots\ldots\ldots\ldots, (a_m, b_m)$$

where a_i indicates m dimensional vector and b_i will be either of -1 or 1 representing the output class to which the sample belongs. The objective will be to find a hyperplane for which distance between the nearest point and hyperplane can be maximized and the classes can be distinguished using the hyperplane.

3.3 Multi-layer Perceptron

Multi-layer perceptron is a feedforward neural network that is used to depict multi-layer neural network or "vanilla" neural network. An elementary MLP will be having an input layer, a hidden layer, and an output layer [38]. It is a supervised learning approach that used back propagation to optimize the random weights which are attached to each hidden layer. In order to distinguish the data, which might not be able to separable using algorithms like SVM and random forest, an activation function is attached to the hidden layer, which is mainly sigmoid activation:

$$f(x) = \frac{1}{1 + e^x}$$

4 System Architecture

The database was created with a collection of 6603 voice recordings from both men and women in nearly equal ratio. This database classified the voice as female or male based on voice and speech acoustic properties. The recordings were done with or without the use of a recorder in both open and closed environment. Each voice sample has been stored with PCM header in .wav format including 3315 male recordings and 3288 female recordings. Further, considering the less amount of existing data, the analysis has further been performed using noise augmentation on both the male and female data such that there exists acoustic mismatch alongside

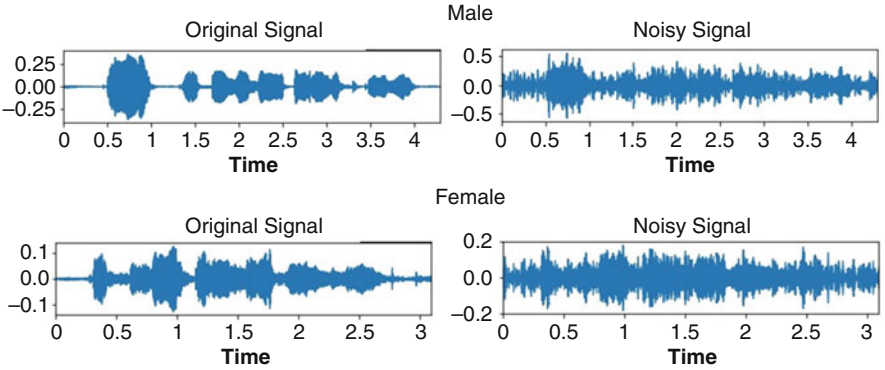

Fig. 1 Visual representation of male and female audio waveform under clean and noisy conditions

variation due to environmental conditions as shown in Fig. 1. Therefore, three different voices—Volvo, Factory, and Babble—from standard NOISEX-92 database [39] has been injected into original dataset at random value which contained much noisy SNR values ranging from −5 dB to 5 dB.

Next, the major focus is on the acoustic feature analysis for the evaluation of the classification performance. The 20 acoustic features—mean frequency (meanfreq), standard deviation corresponding to frequencies (sd), median frequency (median), first quantile (Q25), third quantile (Q75), interquartile range (IQR), skewness (skew), kurtosis (kurt), spectral entropy (sp. ent), spectral flatness (sfm), mode frequency (mode), frequency centroid (centroid), average measure of fundamental frequency (meanfun), minimum measure of fundamental frequency (minfun), maximum measure of fundamental frequency (maxfun), average measure of dominant frequency (meandom), minimum measure of dominant frequency (mindom), maximum measure of dominant frequency (maxdom), range dominant frequency's range across signal (dfrange), and modulation index (modindx) corresponding to male (M) and female (F)—have been extracted using inbuilt R library packages for clean data and noise-augmented data as detailed in Fig. 2a, b respectively.

Perhaps the best and most common machine learning algorithms for classification challenges have been found to be supervised classifiers including random forest, SVM, and MLP. Therefore, the differentiation values of three classification model algorithms utilizing these three classifiers on both clean and noisy datasets as shown in the block diagram in Fig. 3 are being experimented using all 20 features together and three most significant features distinctively.

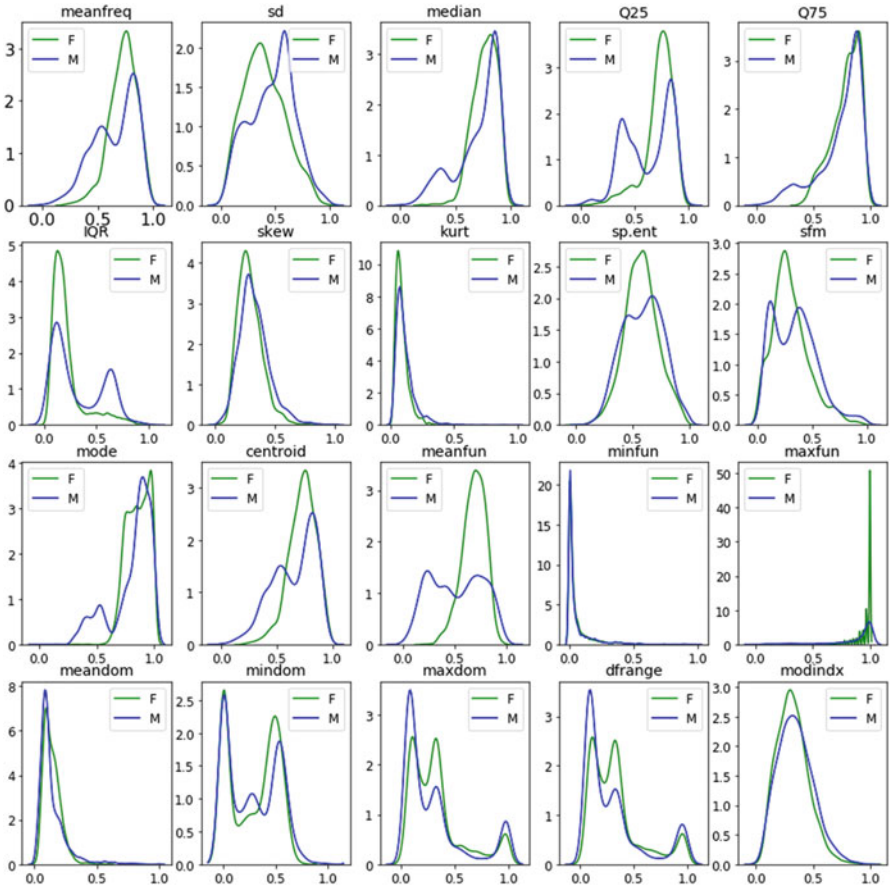

Fig. 2 (**a**) Visualization of acoustic features extracted on clean male and female audio dataset. (**b**) Visualization of acoustic features extracted on noise-augmented male and female audio dataset

5 Results and Discussions

In this section, we address a set of experiments to select the optimal feature parameter model performance for gender recognition from native voice clean dataset. Additionally, the augmented dataset including both noisy and clean dataset has been presented against the classification scheme with an objective of testing the performance of semi-supervised model under degraded conditions.

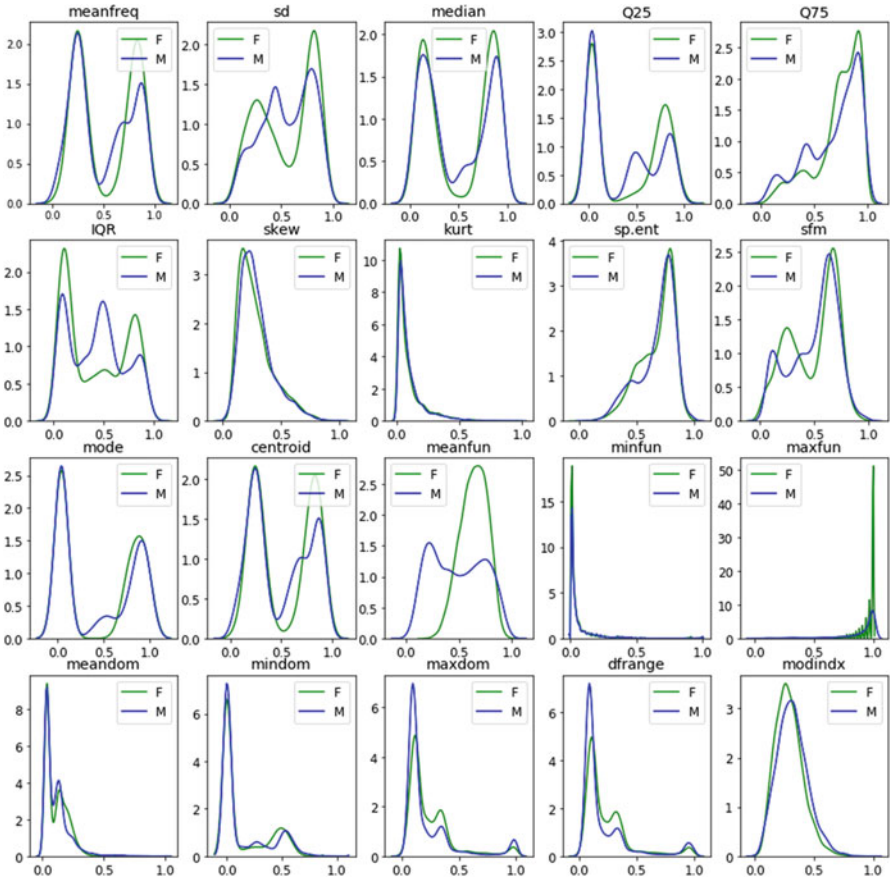

Fig. 2 (continued)

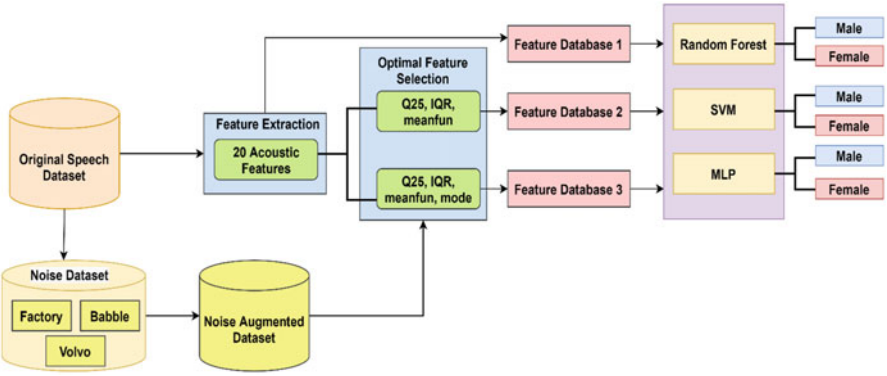

Fig. 3 Block diagram of the proposed gender classification system through optimal acoustic feature selection using noise augmentation

5.1 Performance Evaluation on Clean Dataset

It can be noted that from the outset of extracted features corresponding to the clean dataset as described in Fig. 2a, the three most significant extracted features comparing both male and female audios have been found to be Q25, IQR, and meanfun. Therefore, the comparative analysis for semi-supervised algorithms has been experimented based on the two types of feature selections: one with 20 features all together and the other with three distinctive features as shown in Table 1. In first set of experiments, 20 features resulted into an improved performance than that of ideal selection of three acoustic features. In, order to identify the likelihood of the certain part combinations of all features corresponding to audio set is performed. In addition, given the nonlinear data set corresponding to audio signals, SVM has done no better than the classification method of random forest. However, better performance on the radial basis function (rbf) kernel has been identified with an ideal selection of features with an accuracy of 82.04% over 81.92% with 20 features. Furthermore, 87.28% accuracy utilizing three optimally selected features in case of MLP has outperformed both SVM and random forest classification techniques with an overall RI of 6.54% in case of clean audio dataset.

5.2 Performance Evaluation on Noise-Augmented Dataset

It can be noted that from the outset of extracted features corresponding to the noise-augmented dataset as described in Fig. 2b, the four most significant extracted features comparing both male and female noise-augmented audio sets have been found to be Q25, IQR, mode, and meanfun. Thus, based on baseline results on three preselected optimal feature selections, further experiments on the noise-augmented dataset were conducted with these four most important acoustic features as shown in Table 2. The same spectrum of performances for both random forest and SVM classification techniques is very evident even in case of noisy data. However,

Table 1 Performance evaluation of classification algorithms on clean male and female dataset

	Accuracy (%)	
	20 features	3 features
Random forest	83.04	80.86
SVM	81.92	82.04
MLP	85.56	87.28

Table 2 Performance evaluation of classification algorithms on clean and noise-augmented dataset

	Noise	Accuracy (%)	
		3 features	4 features
Random forest	Yes	86.59	83.28
SVM	Yes	81.80	83.46
MLP	Yes	90.52	92.58
	No	87.28	87.42

random forest classification technique on noise-augmented dataset with 86.59% in Table 2 has outperformed MLP with 85.56% accuracy as in Table 1 utilizing 20 features. Furthermore, two more experiments on MLP classifier with or without noise have shown the relevance of mode frequency parameter such that the classifier utilizing four acoustic features has outperformed the classifier utilizing three acoustic features with an RI of 0.16% on clean dataset and an RI of 2.27% on noise-augmented dataset. Hence, the overall RI of 8.21% in comparison to baseline system has resulted in the development of adequate model classification system for male and female voice.

6 Conclusion

Performing audio analysis can become strenuous while selecting adequate features that can help resolving the cause. Out of numerous features explored, the study found Q25, IQR, and meanfun were able to draw accurate distinction between male and female speakers. Augmentation was applied for creating a noise-robust model alongside adding variability to dataset. After augmenting the dataset, the contour analysis was performed, and this time, mode frequency feature was also included for training of the model and yielded out better performance. MLP outperformed random forest and SVM algorithm, and 8.21% of RI was observed. Using noise-augmented dataset, selection of four features yielded an RI of 6.07%. The research presents opportunity to explore further permutation and combination of feature alongside increasing the corpus. Also, it opens the doors for extending the proposed system for other research areas like age group detection and native and non-native speaker detection.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. Z. Zhang, Mechanics of human voice production and control. J. Acoust. Soc. Am. **140**(4), 2614–2635 (2016). https://doi.org/10.1121/1.4964509
2. J.A. Gómez-García, L. Moro-Velázquez, J.D. Arias-Londoño, J.I. Godino-Llorente, On the design of automatic voice condition analysis systems. Part III: Review of acoustic modelling strategies. Biomed. Signal Process. Contr. **66**, 102049 (2021). https://doi.org/10.1016/j.bspc.2020.102049
3. B. Delgutte, N.Y. Kiang, Speech coding in the auditory nerve: I. vowel-like sounds. J. Acoust. Soc. Am. **75**(3), 866–878 (1984). https://doi.org/10.1121/1.390596
4. D.G. Sinex, C.D. Geisler, Responses of auditory-nerve fibers to consonant–vowel syllables. J. Acoust. Soc. Am. **73**(2), 602–615 (1983). https://doi.org/10.1121/1.389007
5. E.D. Young, M.B. Sachs, Representation of steady-state vowels in the temporal aspects of the discharge patterns of populations of auditory-nerve fibers. J. Acoust. Soc. Am. **66**(5), 1381–1403 (1979). https://doi.org/10.1121/1.383532

6. J.K. Bizley, K.M. Walker, Sensitivity and selectivity of neurons in auditory cortex to the pitch, timbre, and location of sounds. Neuroscientist **16**(4), 453–469 (2010). https://doi.org/10.1177/1073858410371009
7. Hermansky H, Sharma S (1999) Temporal Patterns (TRAPS) in ASR of Noisy Speech. In 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258) (Vol. 1, pp. 289–292). IEEE Phoenix, AZ
8. R. Hu, V. Krishnan, D.V. Anderson, Speech Bandwidth Extension by Improved Codebook Mapping Towards Increased Phonetic Classification, in *Ninth European Conference on Speech Communication and Technology*, (Interspeech, Lisbon, 2005)
9. M. Koo, J. Jeon, H. Moon, M.W. Suh, J.H. Lee, S.H. Oh, M.K. Park, Effects of noise and serial position on free recall of spoken words and pupil dilation during encoding in Normal-hearing adults. Brain Sci. **11**(2), 277 (2021). https://doi.org/10.3390/brainsci11020277
10. M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015). https://doi.org/10.1126/science.aaa8415
11. J. Kaur, A. Singh, V. Kadyan, Automatic speech recognition system for tonal languages: State-of-the-art survey. Arch. Comput. Method. Eng., 1–30 (2020a). https://doi.org/10.1007/s11831-020-09414-4
12. M.H. Moattar, M.M. Homayounpour, A review on speaker diarization systems and approaches. Speech Comm. **54**(10), 1065–1103 (2012). https://doi.org/10.1016/j.specom.2012.05.002
13. D. Raj, P. Denisov, Z. Chen, H. Erdogan, Z. Huang, M. He, et al., Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis. arXiv preprint **arXiv**, 2011.02014 (2020)
14. S.I. Levitan, T. Mishra, S. Bangalore, Automatic Identification of Gender from Speech, in *Proceeding of Speech Prosody*, (Semantic Scholar, 2016), pp. 84–88
15. P. Bawa, V. Kadyan, Noise robust in-domain children speech enhancement for automatic Punjabi recognition system under mismatched conditions. Appl. Acoust. **175**, 107810 (2021). https://doi.org/10.1016/j.apacoust.2020.107810
16. P. Sarma, S.K. Sarma, Syllable based approach for text to speech synthesis of Assamese language: A review. J. Phys. Conf. Series **1706**(1), 012168 (2020) IOP Publishing
17. A. Singh, V. Kadyan, M. Kumar, N. Bassan, ASRoIL: A comprehensive survey for automatic speech recognition of Indian languages. Artif. Intell. Rev., 1–32 (2019). https://doi.org/10.1007/s10462-019-09775-8
18. Y. Kumar, N. Singh, M. Kumar, A. Singh, AutoSSR: An efficient approach for automatic spontaneous speech recognition model for the Punjabi language. Soft. Comput. **25**(2), 1617–1630 (2021). https://doi.org/10.1007/s00500-020-05248-1
19. S. Thomas, M.L. Seltzer, K. Church, H. Hermansky, Deep Neural Network Features and Semi-supervised Training for Low Resource Speech Recognition, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, (IEEE, 2013), pp. 6704–6708. https://doi.org/10.1109/ICASSP.2013.6638959
20. M. Araya-Salas, G. Smith-Vidaurre, warbleR: An R package to streamline analysis of animal acoustic signals. Methods Ecol. Evol. .. **8**(2), 184–191 (2017). https://doi.org/10.1111/2041-210X.12624
21. Y. Zhan, X. Yuan, Audio Post-processing Detection and Identification based on Audio Features, in *2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, (IEEE, Ningbo, China, 2017), pp. 154–158. https://doi.org/10.1109/ICWAPR.2017.8076681
22. G. Li, Y. Liu, The Analysis on the Acoustic Parameters of Distinctive Features for Mandarin Vowels, in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, (IEEE, Shanghai, China, 2017), pp. 1–5. https://doi.org/10.1109/CISP-BMEI.2017.8302104

23. Y. Singh, A. Pillay, E. Jembere, Features of Speech Audio for Accent Recognition, in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, (IEEE, Durban, South Africa, 2020), pp. 1–6. https://doi.org/10.1109/icABCD49160.2020.9183893
24. R. Hidayati, I.K.E. Purnama, M.H. Purnomo, The Extraction of Acoustic Features of Infant Cry for Emotion Detection Based on Pitch and Formants, in *International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009*, (IEEE, Bandung, Indonesia, 2009), pp. 1–5. https://doi.org/10.1109/ICICI-BME.2009.5417242
25. S. Poornima, N. Sripriya, B. Vijayalakshmi, P. Vishnupriya, Attendance Monitoring System Using Facial Recognition with Audio Output and Gender Classification, in *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, (IEEE, Chennai, India, 2017), pp. 1–5. https://doi.org/10.1109/ICCCSP.2017.7944103
26. D. Stewart, H. Wang, J. Shen, P. Miller, Investigations into the Robustness of Audio-Visual Gender Classification to Background Noise and Illumination Effects, in *2009 Digital Image Computing: Techniques and Applications*, (IEEE, Melbourne, VIC, 2009), pp. 168–174. https://doi.org/10.1109/DICTA.2009.34
27. Y.M. Zeng, Z.Y. Wu, T. Falk, W.Y. Chan, Robust GMM based Gender Classification Using Pitch and RASTA-PLP Parameters of Speech, in *2006 International Conference on Machine Learning and Cybernetics*, (IEEE, Dalian, China, 2006), pp. 3376–3379. https://doi.org/10.1109/ICMLC.2006.258497
28. A. Copiaco, C. Ritz, N. Abdulaziz, S. Fasciani, Identifying Optimal Features for Multi-channel Acoustic Scene Classification, in *2019 2nd International Conference on Signal Processing and Information Security (ICSPIS)*, (IEEE, Dubai, 2019), pp. 1–4. https://doi.org/10.1109/ICSPIS48135.2019.9045907
29. P. Gupta, S. Goel, A. Purwar, A Stacked Technique for Gender Recognition Through Voice, in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, (IEEE, Noida, India, 2018), pp. 1–3. https://doi.org/10.1109/IC3.2018.8530520
30. H. Harb, L. Chen, Gender Identification Using a General Audio Classifier, in *2003 International Conference on Multimedia and Expo.ICME'03.Proceedings (Cat.No. 03TH8698)*, vol. 2, (IEEE, Baltimore, MD, 2003), p. II-733. https://doi.org/10.1109/ICME.2003.1221721
31. T.W. Sun, End-to-end speech emotion recognition with gender information. IEEE Access **8**, 152423–152438 (2020). https://doi.org/10.1109/ACCESS.2020.3017462
32. E. Tanuar, E. Abdurachman, F.L. Gaol, Analysis of Gender Identification in Bahasa Indonesia using Supervised Machine Learning Algorithm, in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, (IEEE, Yogyakarta, Indonesia, 2020), pp. 421–424. https://doi.org/10.1109/ICOIACT50329.2020.9332145
33. G.R. Nitisara, S. Suyanto, K.N. Ramadhani, Speech Age-Gender Classification Using Long Short-Term Memory, in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, (IEEE, Yogyakarta, Indonesia, 2020), pp. 358–361. https://doi.org/10.1109/ICOIACT50329.2020.9331995
34. R.D. Alamsyah, S. Suyanto, Speech Gender Classification Using Bidirectional Long Short Term Memory, in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, (IEEE, Yogyakarta, Indonesia, 2020), pp. 646–649. https://doi.org/10.1109/ISRITI51436.2020.9315380
35. S.A. Kokatnoor, B. Krishnan, Twitter Hate Speech Detection using Stacked Weighted Ensemble (SWE) Model, in *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, (IEEE, Bangalore, India, 2020), pp. 87–92. https://doi.org/10.1109/ICRCICN50933.2020.9296199

36. E.E. Kalaycı, B. Doğan, Gender Recognition by Using Acoustic Features of Sound With Deep Learning and Data Mining Methods, in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, (IEEE, Istanbul, Turkey, 2020), pp. 1–4. https://doi.org/10.1109/ASYU50717.2020.9259824
37. P. Mishra, R. Sharma, Gender Differentiated Convolutional Neural Networks for Speech Emotion Recognition, in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, (IEEE, Brno, Czech Republic, 2020), pp. 142–148. https://doi.org/10.1109/ICUMT51630.2020.9222412
38. V. Kadyan, S. Bala, P. Bawa, Training augmentation with TANDEM acoustic modelling in Punjabi adult speech recognition system. Int. J. Speech Technol. **24**, 473–481 (2021). https://doi.org/10.1007/s10772-021-09797-0
39. A. Varga, H.J. Steeneken, Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. Speech Comm. **12**(3), 247–251 (1993). https://doi.org/10.1016/0167-6393(93)90095-3

Index

V. Kadyan et al. (eds.), *Deep Learning Approaches for Spoken and Natural Language Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-030-79778-2