A Review of Textual and Voice Processing Algorithms in the Field of Natural Language Processing

Matt Bowden
Computer Science and Engineering, Australian National University, Australia.
mat.den@outlook.com

Correspondence should be addressed to Matt Bowden : mat.den@outlook.com

Abstract – Currently, there is a significant focus on natural language processing (NLP) within academic circles. As one of the initial domains of inquiry in the domain of machine learning, it has been utilized in a variety of significant sub-disciplines, such as text processing, speech recognition, and machine translation. Natural language processing has contributed to notable progress in computing and artificial intelligence. The recurrent neural network serves as a fundamental component for numerous techniques in domain of NLP. The present article conducts a comprehensive evaluation of various algorithms for processing textual and voice data, accompanied by illustrative instances of their functionality. Various algorithmic outcomes exhibit the advancements achieved in this field during the preceding decade. Our endeavor involved the classification of algorithms based on their respective types and expounding on the scope for future research in this domain. Furthermore, the study elucidates the potential applications of these heterogeneous algorithms and also evaluates the disparities among them through an analysis of the findings. Despite the fact that natural language processing has not yet achieved its ultimate objective of flawlessness, it is plausible that with sufficient exertion, the field will eventually attain it. Currently, a wide variety of artificial intelligence systems use natural language processing algorithms to comprehend human-spoken directions.

Keywords – Natural Language Processing, Machine Learning, Speech Recognition, Machine Translation, Text Processing.

I. INTRODUCTION

Once voice recognition technology achieves 99% accuracy, it is expected to become a primary mode of human-computer interaction [1]. The hypothesis posits that a marginal variance of 4% in precision is the determining factor between being perceived as significantly bothersome and untrustworthy versus being exceedingly beneficial. The application of Deep Learning has enabled us to achieve the pinnacle of success. The rapid advancement of artificial intelligence has made it a widely discussed subject in contemporary discourse. In broad terms, artificial intelligence pertains to a software application that is capable of performing cognitive tasks in a manner that emulates human intelligence. Frequently, this entails the machine emulating the human to accomplish the task in their nonattendance, and on occasion, it may even execute it with greater efficiency than the human. Machine learning is one of the various forms of artificial intelligence.

Machine learning is an approach to augmenting artificial intelligence that leverages learning approaches and the evaluation of diverse forms of data. Neural networks and deep learning represent a distinct category of machine learning. Deep learning algorithms enhance machine learning by repeatedly applying an algorithm to multiple data sets, thereby increasing the machine's knowledge through the results obtained. The utilization of machine learning and computational linguistics is prevalent in the domain of natural language processing (NLP) [2] that is a crucial subfield of computer science. The principal objective of this field of inquiry is to optimize and enhance the interaction between humans and computers. A system capable of comprehending and interpreting human language has been created. The domain of NLP concerns the development of computer programs that are capable of communicating and performing tasks in a manner that is intelligible to humans.

As per Jin, Li, and Tang's [3] assertion, multimodal learning involves the amalgamation of data from multiple sources and modalities to construct a more comprehensive representation. Due to the fact that solely one bit of the encoded word vector is set to 1, the distance between words in the vector remains unaltered following one-hot encoding. As a result, the encoded word vector fails to precisely reflect the correlation between words. The integration of multiple modes of information, commonly referred to as multimodal fusion, is a significant area of focus within the realm of multimodal learning. Multimodal fusion refers to the analytical and classificatory approach that involves the integration of data obtained from multiple sources or modalities. Subsequently, a classifier is trained utilizing the merged feature vectors generated by the feature layer fusion technique. This technique involves the direct implementation of dot-product or point-

sum operations, or even simpler stitching, to multiple distinct feature vectors. One of the advantages is its ease of deployment and ability to offer concurrent access to all pertinent multimodal components.

Integrating all the classifiers into the decision layer, while considering the correlation across modalities, is a challenging task. However, the decision layer has the ability to select the most effective feature classifier and extractor for every form of modality based on unique attributes. **Fig 1** illustrates the utilization of a test set for the purpose of validating the training model and evaluating the comparative advantages of the model algorithms.
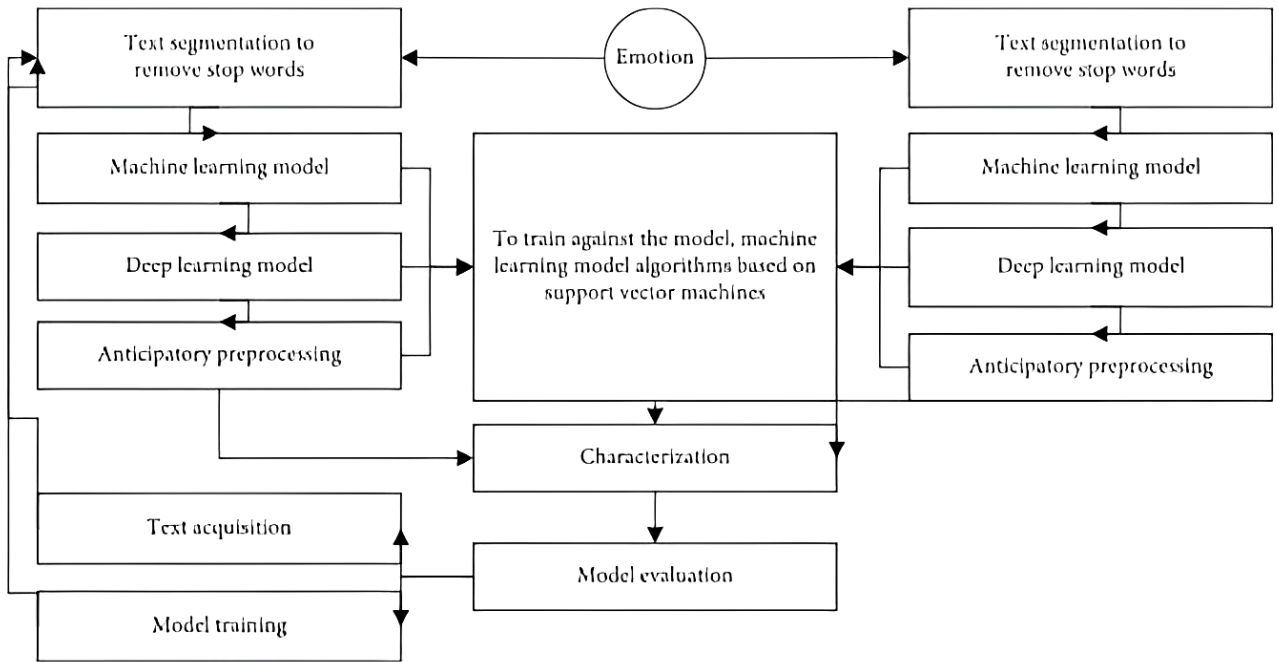

Fig 1. A Schematic Representation of the Natural Language Processing

The significance of natural language processing lies in its capacity to develop models and procedures that can receive input in the form of speech, text, or a combination of both, and subsequently apply an algorithmic transformation within a computer system. The output of a natural language processing (NLP) system can be subjected to processing using diverse inputs such as speech, text, and images. Both verbal and nonverbal forms of communication include spoken words, vocalizations, and written text. This article will discuss several algorithms that have been developed to enhance the efficiency of textual language processing.

The proposed methodology involves utilizing various models such as the template-based approach with automatic text summarization, sequence-2-sequence model, named-entity identification model, long-short-term memory, user preference graph model, feature-based sentence extraction, and word-embedding model, which employs fuzzy inference protocols. Similarly, it is feasible to perform language processing tasks when the input is in spoken form. Numerous algorithms have been developed to serve this objective; however, the most prominent ones include Acoustic Modeling and Word Recognizability. This text discusses several machine translation techniques, including Neural Machine Translation, Google Neural Machine Translation, Connectionist Temporal Classification, and Phase-oriented Machine Translation.

This survey article discusses the evolution of natural language processing algorithms and models, as well as recent advancements in the field. We provide you a high-level understanding of all of the aforementioned algorithms, including how they function, how efficient they are, and the various ways in which they can be used to improve human life. We completed this review article while employed by the SVKM's NMIMS shirpur Computer Science department. The remainder of the article is organized as follows: Section II presents a review of algorithms, models and approaches to text processing and voice processing. Section III focuses on the applications of natural language processing. Section IV review the concept of feature-oriented sentence extraction based on fuzzy inference. In Section V, a review of template-oriented approaches for automated text summarization is provided. Finally, Section VI presents concluding remarks to the article.

II.   ALGORITHMS, MODELS AND APPROACH TO PROBLEM

Text processing approaches
Long Short-Term Memory
The inception of the Long Short-Term Memory (LSTM) neural network was recommended by Schimidhuber and Hochreiter in 1997 [4]. LSTM can be conceptualized as an enhanced iteration of the Recurrent Neural Network (RNN). To effectively address the task of sequence learning, it is imperative to retain information from prior inputs. The challenges of the vanishing gradient and explosion gradient phenomena during prolonged training served as the impetus for the initial conception and subsequent enhancement of LSTM. Due to the presence of the memory cell, the LSTM is capable of retaining inaccurate values and sustaining the gradient flow. Consequently, the problem of disappearance has been

resolved, enabling the extraction of data from sequences that cover a significant number of time steps. LSTM has exhibited superior accuracy compared to other RNN architectures due to its successful resolution of the vanishing gradient issue.

The distinguishing feature of the LSTM from RNN is attributed to the presence of memory gates, such as the input gate, output gate, and the forget gate. In the process of learning, the memory cell undergoes a mechanism wherein it eliminates previously stored information to accommodate more recent and pertinent data. The implementation of the forget gate mechanism is a proposed solution aimed at mitigating this concern. The forget gate is capable of either overwriting or retaining data by modifying the value stored in a memory cell through the addition of a binary digit, either zero or one. In the LSTM architecture, the input gate or write gate is incorporated when it is necessary to preserve a consistent value throughout various stages of the memory cell. The matter pertaining to the competition between memories is additionally tackled through the incorporation of an output gate. A fundamental prerequisite for articulating the LSTM neural network is a comprehensive understanding of the memory gates. The backpropagation (BP) technique is employed in the training phase of the LSTM to forecast wind velocity. Given that the LSTM is employed for the purpose of time series analysis in the context of wind speed prediction, it is imperative that the error pertaining to the complete series be propagated backwards. Consequently, it is also plausible to designate the error propagation of LSTM as Back Propagation Through Time (BPTT) [5]. It is evident that the current cell's state in an LSTM model is impacted by the state of the preceding cell during training.

A recurrent neural network (RNN) is capable of retaining information and subsequently utilizing it for processing activities. RNNs are capable of transmitting information across sequential time steps, rendering them a broader category of neural networks than feedforward neural networks. The models they represent exhibit a wide range of diversity and possess nearly boundless computational capabilities. Xiong, Zhang, Wang, and Yan [6] demonstrated that a finite recurrent neural network with sigmoidal activation functions can emulate a global Turing machine. In recent years, there has been a growing popularity of recurrent neural networks due to their exceptional performance in tasks that necessitate non-independent sequences of points as input and/or output.
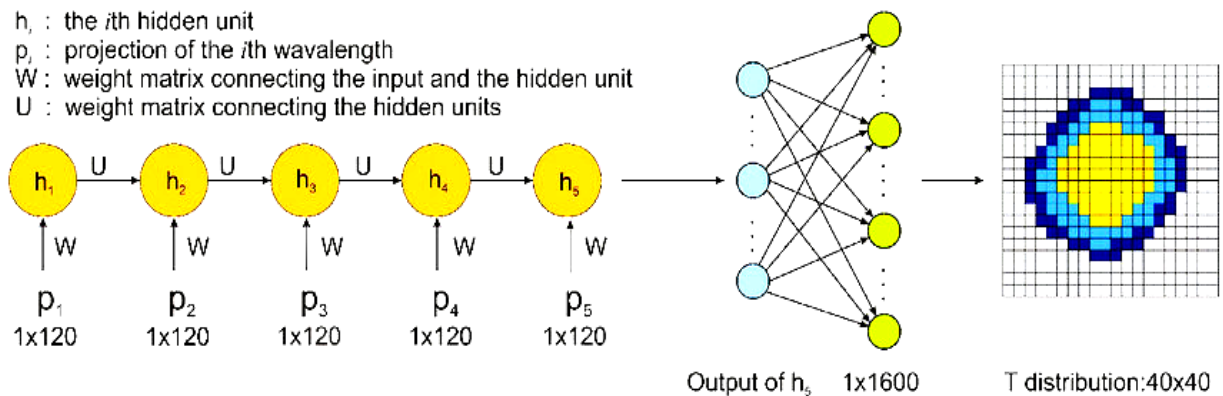

Fig. 2: A Representation of A Typical Recurrent Network

Analogous to a feedforward neural network, the process of activation propagation occurs through interconnected edges that persist over consecutive time steps, denoted as t. The node j' at time t is characterized as j'(t) and is connected to the node j' at time t+1, characterized as j'(t+1), through a dotted line (see **Fig 2**). The third figure illustrates a rudimentary recurrent neural network. The visual representation of the network's dynamics across multiple time steps can be obtained by unfolding the network. The aforementioned depiction enables us to perceive the model as a profound network with a layer assigned to each time step, and weight sharing across iterations, rather than a cyclic model. The network that has been unfolded can be subjected to training through the utilization of backpropagation across multiple timesteps, as is apparent. The Backpropagation through time (BPTT) algorithm was initially devised in 1990.

LSTM recurrent neural networks possess the ability to retain input information over a prolonged duration and make predictions regarding output. The aforementioned is a mechanism utilized to instruct computers novel proficiencies through the utilization of input data sets. The model is widely utilized in the domain of NLP within the realm of machine learning. As the process of learning advances, the previously acquired values are retained without any alteration. While the LSTM model lacks the ability to alter the input data, it is capable of acquiring knowledge from it through the computation of the input's frequency with respect to the incidence of an event. The initial stage of the LSTM process involves the elimination of the input data. The forget gate layer is responsible for determining the extent of memory retention, where a binary value of 0 signifies complete forgetting and a binary value of 1 signifies complete retention. The subsequent stage, ascertained by the input gate layer, encodes and preserves the cell's state. Upon the amalgamation of the input values with the newly formed vector of candidate values produced by the hyperbolic tangent layer, the state undergoes an update. Peephole connections are an auxiliary mechanism present in certain Long Short-Term Memory (LSTM) models, which facilitate the gate's ability to inspect the cell's state prior to discarding any data.

Seq2Seq model

The process of performing conversions using sequence-to-sequence (seq2seq) models is underway [7]. To address the challenge of the vanishing gradient problem, a recurrent neural network (RNN) is utilized, typically in the form of an

LSTM or GRU. The context of each item is determined by the outcome of the operation that precedes it. The system comprises a singular decoder network and a singular encoder network. The process of encoding involves the conversion of individual data points into a latent vector that encapsulates not only the data point itself, but also its contextual surroundings. The decoder utilizes the preceding output as an input to transform the vector into the corresponding output item. Instances of optimizations have been shown in **Table 1**.

Table 1: Instances of seq2seq Optimization	
Attention	The decoder receives a solitary vector comprising the entire context as its input. The attention mechanism enables the decoder to selectively examine the input sequence.
Beam Search	Rather than opting for a solitary output (word), a hierarchical assemblage of probable alternatives is retained through the execution of a Softmax function on the array of attention scores. Compute the mean of the encoder's states while considering the allocation of attention.
Bucketing	The incorporation of zero-padding technique facilitates the handling of sequences with diverse lengths in both input and output. Nonetheless, in cases where the input consists of only three items and the sequence length is one hundred, significant storage capacity is forfeited. It is possible to specify the lengths of input and output independently for each bucket.

The cross-entropy loss function is commonly employed in numerous training scenarios. It operates by imposing a penalty on an output in the event that its subsequent probability is less than 1. The conventional seq2seq architecture comprises of a pair of recurrent neural networks, namely an encoder and a decoder.
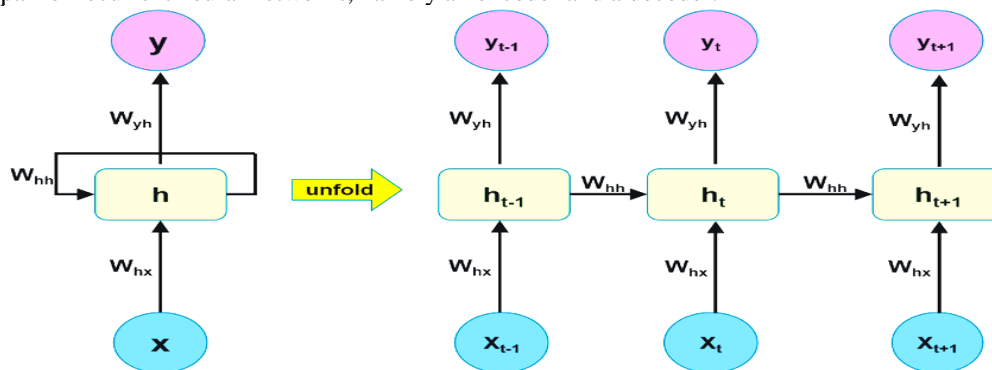

Fig 3. The Structure of the Recurrent Neural Network

Fig 3 comprises individual RNN cells, which are predominantly LSTM-based RNNs. Within the paradigm in **Fig 4**, every input is transformed into a vector of predetermined dimensions, and the decoder is employed to restore the initial input.
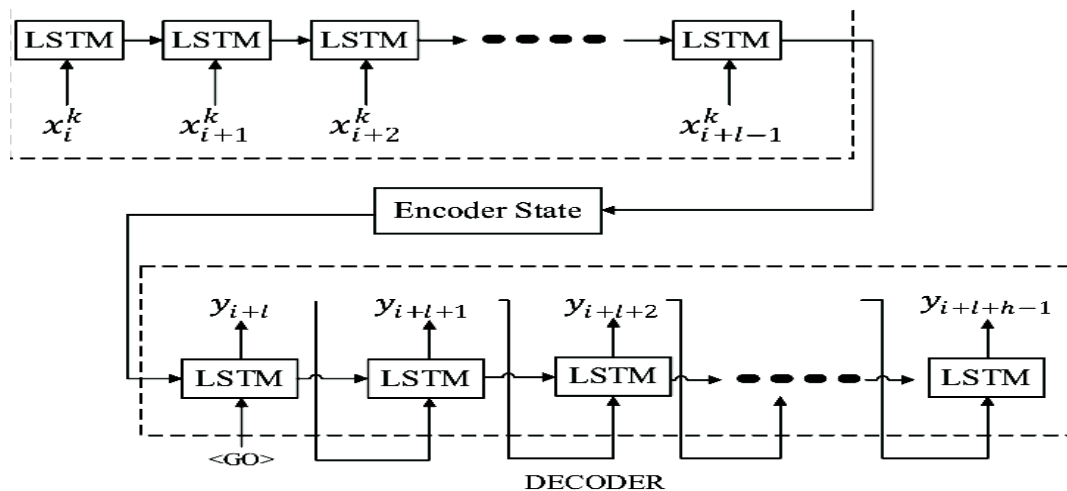

Fig 4. The Operation and Structure of the Encoder-Decoder

To facilitate the recognition of appropriate grammatical syntax by the model, a lexicon is initially formulated and generated through embedding. The process of scrutinizing the lexicon is undertaken to discern occurrences of lexical items and classify them into one of three categories: highly prevalent, rare, or distinctive. Subsequently, they undergo a transformation into identification numbers (IDs). The proposal for the encoded response is decoded and generated utilizing identification numbers.

Named Entity Recognition Model
The process of identifying significant entities and classifying them based on their respective categories is referred to as named entity recognition. The data sets utilized for analysis can be categorized as either textual or acoustic. Named Entity Recognition (NER) models [8] are employed to detect and classify entities such as names, locations, and individuals, among others. The NER model comprises of a two-stage procedure. The initial stage of a Named Entity Recognition (NER) model involves the segmentation of the given text into smaller units that can be conveniently classified. Tokens are employed to categorize these segments into conventional types, such as proper nouns denoting individuals, locations, and organizations. The utilization of typographical features such as bolding and capitalization is disregarded in academic writing. On March 19th, a student named Mike, who attends New York University, received a grade of 90% in his seminar, during the second iteration of the model. The construction of a user preference graph for the purpose of providing intelligent search recommendations is a concept that can be widely extended to the domains of language and voice processing.

User Preference Graph
To produce a collection of user preferences, the user preference graph is used. When a user repeatedly uses the same set of conjunctions, prepositions, adjectives, and tenses, a preference graph is structured. With the help of this graph, the model can calculate probabilities for the next words in the user's phrases. The words are connected in a structured way, and the resultant graph represents the user's preferences (see **Fig 5**). Individuals showing similar user preference graphs are grouped together in a large-scale application of this approach to permit a wide variety of suggestions. Smart reply, smart typing suggestions, and automatic response systems are just a few examples of how this capability might be implemented into existing technologies.
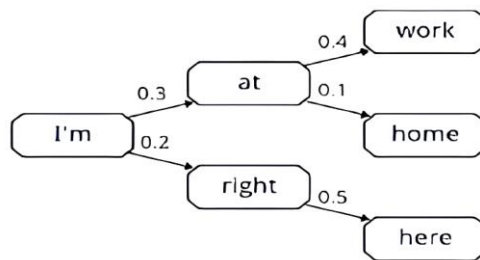

Fig 5. A Typical Example of the User Preference Graph

Word Embedding
The word embedding process [9] is a methodology employed in the domain of NLP, which involves the mapping of words and phrases onto a preference network that is represented as a vector of real numbers. This methodology is a resultant of the process of feature acquisition and linguistic modeling. This refers to a technique utilized to transform textual information and documents into a coded format. A word vector, commonly referred to as a word embedding, is a numeric depiction of a word that can be utilized in a reduced dimensional context. This facilitates the representation of synonymous terms in a uniform manner. They may also produce proximate meanings. A set of fifty values represented in a word vector has the ability to signify fifty unique attributes.

Phrase Based Machine Translation
The Phrase Based Machine Translation (PBMT) [10] is a possible manifestation of the Statistical Machine Translation (SMT). The process of text translation is facilitated through the utilization of predictive modeling techniques. The models are constructed through the utilization of an extensive and disorganized collection of multilingual literary works, or alternatively, instructed with the aid of such a corpus. The figure-oriented translation framework illustrated in **Fig 6** is utilized to obtain the most probable output.
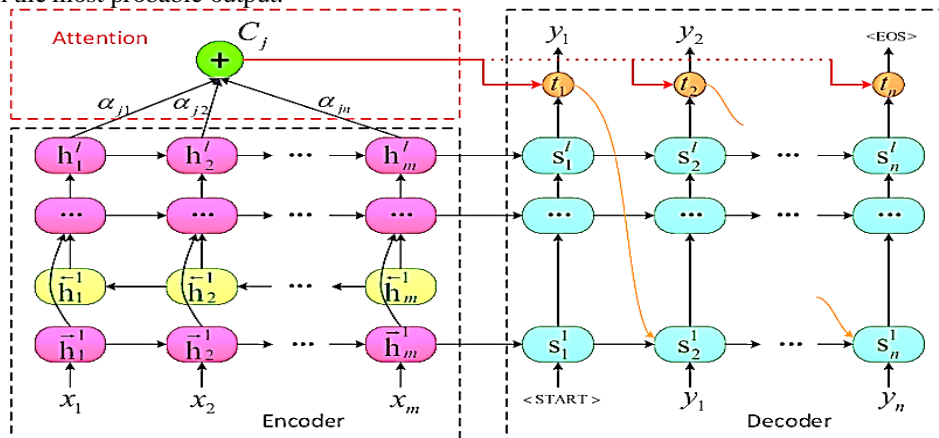

Fig 6. A Representation of the Figure-Oriented Translation Framework for Encoder-Decoder

The PBMT's algorithm is show as follows:

a) Deconstruction of the initial sentences into chunks.

b) Identification of the potential translation for every chunk: In this stage, the utilization of corpora is employed to assess how individuals have translated the given text into sentences that are representative of real-life usage.

c) Generation of all potential sentences and identifying the best one: By utilizing various translation combinations in step b), it is possible to produce over 5000 sentence permutations.

d) Providing potential scores by relating them to training datasets: The present scenario involves a training set that comprises a substantial corpus of written material sourced from multiple origins. The probability value, also known as a likelihood score, is assigned to every conceivable outcome resulting from the previous steps through comparison with the training dataset. Each permutation will be subjected to analysis against our training data set to determine the permutation that produces the highest likelihood score and the most plausible translation of chunks.

PBMT's disadvantage is that it is challenging to structure and maintain. In the event of a new language being considered for inclusion, it is advisable that bilingual corpora be readily accessible. In instances where language combinations are infrequent, there may be a need to make concessions in the process of translation. This presupposes that a complex pipeline is unnecessary for the translation of Gujarati to Georgian. One potential option could involve performing an internal translation from the original language into English, and subsequently translating the English text into Georgian.

Neural Machine Translation
Neural Machine Translation is currently regarded as the most contemporary methodology for Machine Translation. In contrast to Statistical Machine Translation, the translations generated by it exhibit a higher degree of precision. Neural machine translation employs multiple layers of processing to ensure precise translation of the input. NMT has the capability to utilize an algorithmic approach for deducing linguistic rules from statistical models, thereby enabling autonomous acquisition of language. The NMT (Neural Machine Translation) system [11] employs subword units and relies on attentional encoder-decoder technology. The utilization of back-translation from the single-language news corpus has been implemented as supplementary training data to further enhance efficacy. The functionality is effective in both forward and reverse directions. The strengths of NMT reside in its capacity to address and forestall instances of verb ellipsis and alternative verb ordering patterns. This software has the ability to gather English nouns. NMT has demonstrated efficacy in managing grammatical aspects such as phrase structure and articles. NMT's limitations include words that possess multiple potential translations in the German language. The construction of progressive verb tenses poses a similar challenge. Prepositions pose a significant challenge to NMT.

Voice Processing Approaches
Acoustic Modeling
Lexical items are phonetically segmented into their constituent phonemes, and their corresponding phonemic transcriptions can be located at this location. Subsequently, each of the aforementioned auditory stimuli is assigned a specific nomenclature. The linguistic term used to refer to these symbols is phonemes. The statistical representations of individual phonemes in a language, known as AQV presentations, are produced through the utilization of a voice corpus and a specialized training algorithm. The statistical model of this nature is commonly referred to as the Hidden Markov Model (HMM) [12]. Each phoneme is associated with a Hidden Markov Model (HMM). Acoustic modeling confers various advantages, such as promoting proper speech articulation among users, obtaining superior audio recordings using smartphones, and transmitting said vocal data to a server via IP with zero inaccuracies. These models are likely to yield favorable outcomes for users following extensive training with a substantial volume of data. However, this perspective is not shared by many others. The reason for this is that the model was produced by utilizing data from numerous speakers, thereby rendering the outcomes generalizable. Developing customized models for individual users or languages with limited speaker populations is not a financially viable approach. The limitations of the acoustic modeling of the technology represent a significant disadvantage.

Connectionist Temporal Classification
The effective training of recurrent neural networks (RNNs) is largely dependent on the utilization of Connectionist Temporal Classification. The LSTM model represents a specific form of RNN. The significance of time in the context of voice recognition is of utmost importance. The input resembling a phenotype may exhibit temporal variability. To address the phoneme recognition limitations of LSTM, it is imperative to identify strategies for enhancing its performance. Connectionist Temporal Classification (CTC) algorithms are independent of the underlying network topology, as they operate through scoring and output mechanisms. The concept of CTC was initially introduced by Wang and Jang [13]. The process of training a CTC network involves the estimation of the probability of a label, which is determined by the continuous output of the network. Labels are generated sequentially. If the sole dissimilarity between two groupings of designations is their respective magnitudes, then those groupings are deemed to be of equal value. There are numerous variations of indistinguishable labels that can be observed. Consequently, the process of grading is a challenging task.

The approach is presented as:

Transforming sounds to bits

The technical nomenclature for this process is referred to as "sampling." Nyquist's theorem states that the reconstruction of the original signal is possible if it is sampled at rates, which is two times greater in frequency available in signals. Yolanda, Hersyah, and Pratama [14] have demonstrated that a sample frequency of approximately 16,000 samples in every second is best for the purpose of voice recognition. Subsequently, a sequence of numerical values will be presented, denoting the magnitude of the acoustic waveform at intervals of 1/16000 seconds. Directly inputting the sampled data into a neural network may prove to be a challenging task, as it would require significant effort to discern the underlying pattern within the voluminous data set extending the duration of the algorithm's execution. To clarify, the initial phase of processing has been completed.
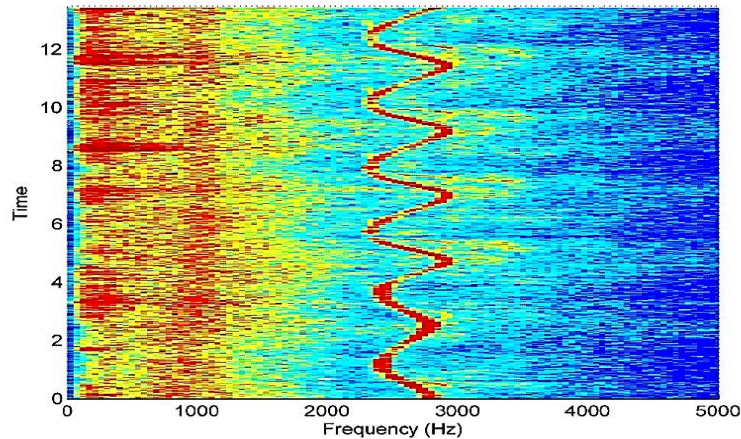
Fig 7. 12ms/5000 Samples/Window

Preprocessing sample dataset

The algorithm's complexity of time is reduced through the implementation of preprocessing techniques. The preprocessing stage involves segmenting the sampled audio into chunks that are 12 milliseconds (ms) in duration. The image presented below displays the initial 12 ms of audio, specifically the initial 5000 samples. The processing of this brief recording is challenging due to the intricate nature of human speech sounds. The speech spectrum encompasses low-frequency sounds, high-frequency speech sounds, and mid-frequency speech sounds. In order to further decrease the complexity of time, the Fourier Transform is employed. By means of the Fourier Transform, the intricate sound wave is decomposed into elementary sound waves. Subsequently, the total amount of energy contained in each entity is computed. The creation of a spectrogram is motivated by the fact that neural networks are better equipped to identify patterns in spectrograms as opposed to raw sound files. The spectrogram in **Fig 7** depicts the sampled data as shown above.

Identifying the features from short sounds

Due to the simplicity of the input, speech processing can easily be accomplished (see **Fig 8**). The deep neural network can be fed with it. The 20-millisecond audio segments serve as inputs. The neural network would endeavor to discern the phoneme for every input. As a recurrent neural network, the current output has an impact on future predictions. For the purpose of clarity, let us assume that the input data pertains to an individual uttering the term "HELLO". Given that the current neural network has successfully identified the sequence "HEL" thus far, it is highly probable that the subsequent sequence produced will be "LO" instead of an arbitrary word like "XYZ". Given that CTC is also involved in managing audio segments of different lengths. The resulting output would be "HHHEE_LL_LLLOOO" in accordance with the given input. The user's text does not contain any recognizable academic language or structure. Therefore, it cannot be rewritten in an academic style without additional context or information. Subsequently, the output will be refined through the elimination of duplicated characters.

The original text "HHHEE_LL_LLLOOO" has been modified to "HE_L_LO" in a manner that conforms to academic writing standards. The original text "HHHUU_LL_LLLOOO" has been modified to "HU_L_LO" in an academic manner. The original text "AAAUU_LL_LLLOOO" has been transformed into "AU_L_LO". Subsequently, we will eliminate any empty spaces. The string "HE_L_LO" is transformed into "HELLO" through a process of character substitution. The word "HU_L_LO" is transformed into "HULLO". The abbreviation AU_L_LO is transformed into the condensed form AULLO. Given that all of the sounds bear resemblance to the utterance "HELLO". The proposed approach involves integrating pronunciation-based predictions with likelihood scores derived from a vast corpus of written text. Therefore, based on the likelihood score, it can be inferred that the probability of "Hello" is higher than the other two options. In order to ensure accurate display of the results, it is necessary to ensure that the output is presented correctly. One limitation of this algorithm is that it may not accurately recognize an input audio file containing the word "HULLO" due to the relatively low frequency of occurrences of this word in the written text database. The algorithm may experience a malfunction in the event that the reader utters words that are not included in the amount of transcribed text.
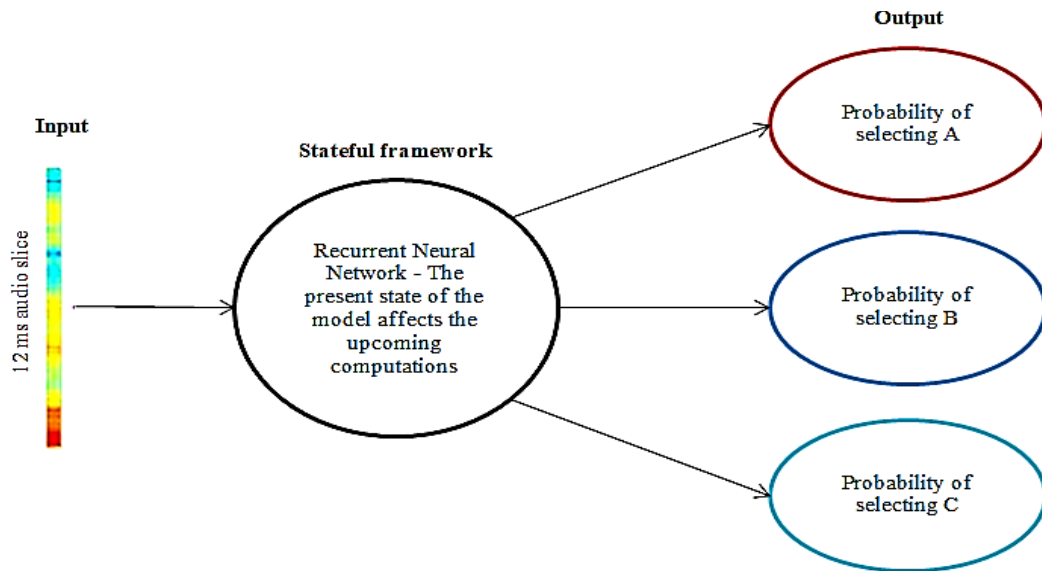
Fig 8: Operation of the Networking System for Speech Processing

III.   APPLICATIONS OF NATURAL LANGUAGE PROCESSING

This discussion pertains to the application of Natural Language Processing in the context of automated text summarization using software. This study aims to analyze and contrast two prominent algorithms utilized for text summarization, with the objective of arriving at a definitive conclusion. Before delving into the algorithm, it is beneficial to obtain a more profound comprehension of automatic text summarization. Automated text summarization is the act of utilizing application to condense big data into a brief and informative text. The aim of this procedure is to facilitate the readers' understanding of the document's contents in a concise and informative manner, thus conserving time and energy.   There are two predominant techniques utilized for the automated condensation of textual content. The two aforementioned procedures are extraction and abstraction.

Text summarization employs extractive techniques to choose a collection of sentences, words, or phrases from the source document for the purpose of generating a summary. Abstractive techniques involve the creation of an internal semantic representation by the algorithm, which is then utilized to implement natural language generation methods. The methodology entails the utilization of artificial intelligence to simulate the cognitive functions of humans in order to produce a cohesive synopsis through the comprehension of the written material within the document. The aforementioned process produces a summary that closely approximates what an individual would extract and present as a summary of written material. The current synopsis comprises innovative linguistic phrases. Thus far, academic inquiries have primarily focused on methods of extraction that can be utilized for the consolidation of collections of images, textual materials, and videos.

IV.   FEATURE-ORIENTED SENTENCE EXTRACTION BASED ON FUZZY INFERENCE

As per the prescribed approach, every expression in the input information is evaluated based on a predetermined set of standards that categorize each assertion into one of three levels: "low," "medium," or "high." The process of assigning values involves the utilization of fuzzy rules, also known as fuzzy logic, comprising a set of 8 rules. The aforementioned regulations employ a conditional structure in the form of IF-THEN statements. When provided with input statement F, the algorithm executes all applicable rules and assigns corresponding values.

Similar to the logical rule that IF (F1 is H) denotes critical importance, the term "high" in this context also conveys the same significance. The aforementioned sequence persists as follows: F2 corresponds to H, F3 corresponds to M, F4 corresponds to H, F5 corresponds to M, F6 corresponds to H, F7 corresponds to H, and F8 corresponds to H. Once the rules have been reviewed, any pertinent statement is incorporated into the summary in the original sequence as presented in the input data. The methodology comprises of four sequential steps that collectively yield a condensed rendition of the input data. The initial step in the process is preprocessing, which is subsequently followed by the extraction of features. The final stage involves the application of Fuzzy logic to assign scores. Sentence construction and word choice are important aspects of effective communication. The output of a given step is transmitted as input to the subsequent step, and this process is repeated iteratively across multiple stages.

The algorithm took into account multiple variables prior to determining the inclusion or exclusion of a specific phrase within the summary. The second variable of the title pertains to the concept of Term-Weighted Average in academic discourse. The third variable to consider in sentence structure is the length of the sentence, followed by its placement in the sentence. The thematic adjective is a descriptive word that is used to convey a particular theme or topic in a text. It is often used to provide additional context and meaning to the subject matter being discussed. The use of thematic adjectives can help to create a more nuanced and sophisticated analysis of a given topic. Subsequently, the software assesses the given text utilizing these attributes and generates an outcome.

V.   TEMPLATE-ORIENTED APPROACHES FOR AUTOMATED TEXT SUMMARIZATION

In contrast to the template-based extraction algorithm, the feature-based algorithm employs a set of fundamental criteria or features to assess the sentences. The sentences are then incorporated into the output summary with minimal modification from their original form in the input data. This approach differs from the former, which utilizes the extracted text to construct a summary that appears more human-like in terms of grammar and structure. The implementation of automatic text summarization through a template-based approach involves a two-stage process, as described by Maylawati, Kumar, and Binti Kasmin [15].

Text pre-processing

This component of the execution encompasses the module depicted in **Table 2**.

Table 2: Text Pre-Processing Modules	
Syntactic analysis	The objective of the syntactic evaluation component is to identify the precise locations within the input document where sentences commence and terminate. The present algorithm regards the period as denoting the end of a sentence. Furthermore, it is noteworthy that any text preceding the concluding punctuation mark is regarded as a self-contained idea.
Tokenizer	Tokenization is the process of segmenting parsed sentences using syntactic analysis modules. A phrase may consist of fragmented components such as words, punctuation marks, and numerals.
Semantic	During the semantic evaluation stage, the interpretation of individual words within a given phrase is derived. Subsequently, each individual noun, verb, adjective, adverb, and other linguistic elements are assigned a corresponding tag. The process of assigning words to their corresponding grammatical categories is referred to as part-of-speech (POS) tagging.
Stop Word Removal	Certain lexical items exhibit higher frequency in natural language texts, but their contribution to the process of meaning extraction is minimal when the entire contextual information of the phrase is considered. The terms referred to as "stop words" are expressly forbidden for usage.
Stemming	Stemming refers to the process of defining the fundamental form of a particular word within a text document. In order to prevent redundancy, stemming is utilized to convert words with equivalent meanings but varying tenses into their fundamental, uncomplicated tense.

Information extraction

The algorithmic segment in question encompasses several distinct modules, as illustrated in **Table 3**.

Table 3: Information Extraction Modules	
Training the dialogue management	The framework necessitates knowledge of significant or indicative terms, designated entities (such as proper nouns denoting individuals, places, and temporal references), and other normative principles through the process of training. The efficacy of the algorithm increases proportionally with the number of training batches it undergoes, as it acquires additional knowledge and expertise.
Knowledge-based discovery	The term "knowledge-based discovery" refers to the procedure of extracting insightful data and archiving it in an unorganized textual format. Consequently, the reduction of the necessity to generate numerous storage structures for the storage of diverse terms of various categories results in a decrease in search time hence enhances the algorithm's overall performance.
Dialogue management	This module is a software system that facilitates interaction between human and machine. This module facilitates the user to generate queries using natural language in a seamless manner. The training model functions as the experiential dataset for the conversation control module.
Template-oriented summarization	Summarization is the act of condensing the significant textual content within a given document or input data into a given format.

The algorithm additionally enables the user to establish templates, which integrate particular provisions for specifying named entities, locations, events, and other relevant information. The user is afforded the opportunity to designate a variable quantity and diverse array of POS patterns.

VI.   CONCLUSION

The field of natural language processing has experienced significant progress due to the confluence of several factors, including the abundance of big data, enhanced computational capabilities, refined algorithms, and an increasing fascination with human-machine interactions. The preceding exposition elucidates that the algorithms of text processing depend on preference graphs and entity-oriented categorization as their foundational elements. Text processing algorithms are employed in various popular applications to offer "smart suggestions" and "smart reply" features, which aim to provide valuable outcomes while minimizing the user's time and effort. Although the task of voice processing remains unresolved, notable progress has been made in this field during the past decade. The utilization of neural and deep learning techniques has been observed to yield enhanced outcomes in the domains of text processing and voice processing. Significant advancements in this domain have been achieved due to the implementation of recently developed algorithms. Recent advancements in algorithms have significantly enhanced their precision, resulting in output accuracy that closely approximates human interpretation. Various forms of artificial intelligence have been developed utilizing voice processing and text processing approaches to evaluate the requirement of users with respect to the classification of input data. As a

result of this, the resultant consequences are superior, and the end-user obtains a more customized and personalized encounter. The amalgamation of multiple text processing techniques with voice processing algorithms yields enhanced outcomes.

Data Availability
No data was used to support this study.

Conflicts of Interests
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding was received to assist with the preparation of this manuscript.

Ethics Approval and Consent to Participate
The research has consent for Ethical Approval and Consent to participate.

Competing Interests
There are no competing interests.

References

[1].   K. Ashok, M. Ashraf, J. Thimmia Raja, M. Z. Hussain, D. K. Singh, and A. Haldorai, "Collaborative analysis of audio-visual speech synthesis with sensor measurements for regulating human–robot interaction," International Journal of System Assurance Engineering and Management, Aug. 2022, doi: 10.1007/s13198-022-01709-y.

[2].   H and A. R, "Artificial Intelligence and Machine Learning for Enterprise Management," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Nov. 2019, doi: 10.1109/icssit46314.2019.8987964.

[3].   L. Jin, Z. Li, and J. Tang, "Deep semantic multimodal hashing network for scalable image-text and video-text retrievals," IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 4, pp. 1838–1851, 2023.

[4].   W. Wang, R. Ma, T. Luo, C. Long, Q. Ye, and D. Chen, "X-12 seasonal adjustment combined with long and short-term memory neural networks for monthly electricity sales forecasting," in 2022 International Symposium on Advances in Informatics, Electronics and Education (ISAIEE), 2022.

[5].   N. Rajput and S. K. Verma, "Speech recognition using the epochwise back propagation through time algorithm," Int. J. Comput. Appl., vol. 95, no. 21, pp. 17–21, 2014.

[6].   J.-J. Xiong, G.-B. Zhang, J.-X. Wang, and T.-H. Yan, "Improved sliding mode control for finite-time synchronization of nonidentical delayed recurrent neural networks," IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 6, pp. 2209–2216, 2020.

[7].   W. Luan, R. Zhang, B. Liu, B. Zhao, and Y. Yu, "Leveraging sequence-to-sequence learning for online non-intrusive load monitoring in edge device," Int. J. Electr. Power Energy Syst., vol. 148, no. 108910, p. 108910, 2023.

[8].   Haldorai and U. Kandaswamy, "Energy Efficient Network Selection for Cognitive Spectrum Handovers," EAI/Springer Innovations in Communication and Computing, pp. 41–64, 2019, doi: 10.1007/978-3-030-15416-5_3.

[9].   H. Cao, T. Zhao, W. Wang, and W. Peng, "Bilingual word embedding fusion for robust unsupervised bilingual lexicon induction," Inf. Fusion, vol. 97, no. 101818, p. 101818, 2023.

[10].   G. Qiu, "Parallel algorithm of hierarchical phrase machine translation based on distributed network memory," Int. J. Inf. Syst. Supply Chain Manag., vol. 15, no. 1, pp. 1–16, 2021.

[11].   Haldorai, A. Ramu, and S. Murugan, "Computing and Communication Systems in Urban Development," Urban Computing, 2019, doi: 10.1007/978-3-030-26013-2.

[12].   V. Markov, V. Rastunkov, A. Deshmukh, D. Fry, and C. Stefanski, "Implementation and learning of quantum hidden Markov models," arXiv [quant-ph], 2022.

[13].   J.-Y. Wang and J.-S. R. Jang, "Training a singing transcription model using connectionist temporal classification loss and cross-entropy loss," IEEE ACM Trans. Audio Speech Lang. Process., vol. 31, pp. 383–396, 2023.

[14].   D. Yolanda, M. H. Hersyah, and R. N. Pratama, "Implementation of Mel frequency cepstrum coefficients and dynamic time warping algorithms on door physical security system using voice recognition pattern," in 2020 International Conference on Information Technology Systems and Innovation (ICITSI), 2020.

[15].   D. S. Maylawati, Y. J. Kumar, and F. Binti Kasmin, "Feature-based approach and sequential pattern mining to enhance quality of Indonesian automatic text summarization," Indones. J. Electr. Eng. Comput. Sci., vol. 30, no. 3, p. 1795, 2023.