

Review of Computer Engineering Research

2024 Vol. 11, No. 4, pp. 155-187

ISSN(e): 2410-9142

ISSN(p): 2412-4281

DOI: 10.18488/76.v11i4.4018

© 2024 Conscientia Beam. All Rights Reserved.




Natural language processing: The current state of the art and challenges

 **Mohammad
Mustafa Taye^{1*}**

¹Software Engineering, Philadelphia University, Amman 19392, Jordan.

¹Email: mtaye@philadelphia.edu.jo

 **Rawan Abulail²**

²Computer Science, Philadelphia University, Amman 19392, Jordan.

²Email: rabilail@philadelphia.edu.jo

 **Belal Al-Ifan³**

^{3,4}Data Science and Artificial Intelligence, Philadelphia University, Amman 19392, Jordan.

³Email: balifan@philadelphia.edu.jo

 **Fadi Alsuhimat⁴**

⁴Email: fshimat@philadelphia.edu.jo



(+ Corresponding author)

ABSTRACT

Article History

Received: 24 October 2024

Revised: 2 December 2024

Accepted: 17 December 2024

Published: 27 December 2024

Keywords

Deep learning
Feature extraction
Formatting
Machine learning
Natural language processing
NLP applications.
Word embedding.

The use of deep learning techniques in natural language processing (NLP) is examined thoroughly in this study with particular attention to tasks where deep learning has been shown to perform very effectively. The primary strategies explored are phrase embedding, function extraction, and textual content cleaning. These are all essential for sorting textual content statistics and files. It appears in important gear like software programs, hardware and extensively used libraries and cutting-edge programs for deep learning in NLP. In NLP, deep learning is turning into a chief fashion, changing many areas and making large modifications in many fields. This paper stresses how deep analyzing techniques have a good-sized-ranging effect and how vital they may be for shifting the world in advance. This paper also discusses how deep learning may assist in solving modern issues and handling challenging and stressful situations in NLP research. Since those methods are getting more popular, it indicates that they're top at handling many NLP responsibilities. The final part of the evaluation talks about the most current makes use of, developing traits and long-term troubles in NLP. It helps practitioners and lecturers determine and use the capabilities of deep learning in the dynamic field of natural language processing with its applicable facts and examples.

Contribution/Originality: This study differs in that it provides a comprehensive analysis of recent deep learning methods in NLP by combining an investigation of theoretical foundations with practical implementations. Unlike earlier research, it focuses on generative models, unsupervised and reinforcement learning approaches, and emerging trends, providing a comprehensive view of NLP's evolving landscape.

1. INTRODUCTION

Natural Language Processing (NLP) is the field that makes a speciality of growing computer programs that efficaciously manage and analyze considerable portions of natural language [1]. NLP is a topic that intersects linguistics, laptop science and artificial intelligence. For a long time, natural language processing (NLP) has been superior to rule-based methodologies, statistical techniques and AI-driven applications, yielding noteworthy effects in domain names along with text categorization, sentiment evaluation, gadget translation, voice popularity, and textual content synthesis [2].

The intricacy of processing unstructured text data such as that produced by the growth of social media platforms, forums and papers makes it difficult to obtain valuable information from these sources. Natural

Language Processing (NLP) significantly helps in obtaining valuable information and insights from such sources. Certain deep learning models have solved some of the most challenging NLP tasks due to recent advancements in computing and easier access to computing resources. The field of NLP develops models, systems and algorithms to improve our ability to understand human language [3].

Indeed, NLP divides into two distinct areas: theoretical (basic) study and practical application. In the first group, it came across wide-ranging issues with expressing the building blocks of language-based systems of varying complexity. Language modeling, morphological analysis, syntactic processing, parsing and semantic analysis are just some examples of these activities. NLP also addresses practical issues such as the translation of text between languages, document summarization, automatic question answering, document classification and document clustering in addition to these theoretical concerns [4].

In both cases, massive neural networks have proven to be more effective than conventional ML algorithms like support vector machines (SVM). The primary benefit of these models is their ease of adoption. They often just need a single end-to-end architecture for training and do not necessitate the customary feature engineering required for a given activity [5].

On the other hand, there is no restriction on training data that deep neural networks may use. However, the spread of neural techniques has been delayed by the scarcity of semantically annotated data which often necessitates specialized human effort for activities linked to the semantic analysis of natural languages.

In recent years, attracting the attention of the machine learning community due to deep learning's improvements, NLP applications have seen a dramatic increase in performance. Newer models have also begun to outperform humans in a variety of tasks such as question answering and lying content detection [6]. Various issues must be resolved such as the computational cost, the reproducibility of results and the lack of interpretability even if recent algorithms are beginning to reach excellent performance on a variety of tasks. Several deep learning and NLP-related literature reviews have appeared in the past few years.

Models of Natural Language Processing (NLP) can be created to translate and interpret various textual elements, including phonology, morphology, grammar, syntax and semantics. NLP can also register character-, word- and phrase-levels and sentence- or document-level language modeling depending on the building elements. Traditional NLP models are constructed using human linguistic expertise to handcraft features. Researchers have begun to broaden models that could decipher entire chunks of textual content without explicitly parsing out the relationships between words inside a sentence rather than using raw textual content directly with the current boom in quit-to-quit education for deep studying fashions [4].

Natural Language Processing (NLP) employs an extensive range of computational strategies grounded in linguistic concepts to routinely examine and constitute human language. Research in natural language processing (NLP) has stepped forward from the time of punch playing cards and batch processing when analyzing a sentence may absorb 7 minutes to the contemporary when tens of millions of webpages may be processed in much less than a 2nd way by search engines like Google and Yahoo [7]. Parsing, POS tagging, gadget translation, and communication systems are just some of the numerous Natural Language Processing (NLP) activities that computer systems can now execute [8, 9].

Deep learning systems and algorithms have already provided significant improvements in computer vision and pattern recognition. Deep models (such as support vector machines and logistic regression) trained on very high-dimensional and sparse features have formed the backbone of NLP-focused machine-learning approaches for decades. Neural networks trained on dense vector representations have outperformed other types of NLP models in recent years. Word embedding by Kaddari et al. [4], Farzindar et al. [10] and Li and Yang's [11] deep learning [12, 13] have spurred this movement. Deep learning makes it possible to automatically learn multi-level feature representations. On the other hand, classic NLP systems that rely on machine learning rely primarily on manually created features. Such hand-made additions are labor-intensive and frequently flawed [13, 14].

The state of the art in machine translation has shifted from phrase-based statistical approaches to neural machine translation which is made of huge deep neural networks and produces better performance. The use of dictionaries, ontologies and rules of syntactic grammar for named entity recognition has also been superseded by recurrent architectures and deep learning models [15, 16].

Some Natural Language Processing (NLP) tasks such as Named Entity Recognition (NER), Semantic Role Labeling (SRL), and Point-of-Sale (POS) tagging have shown that a basic deep learning architecture outperforms most state of the art methods [8, 16, 17]. Several advanced methods based on deep learning were proposed to address difficult natural language processing issues. This article discusses a variety of deep learning models and methodologies, including several that have proven successful in natural language applications such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and recursive neural networks [17]. In addition, we may explore attention mechanisms, memory augmentation methods, and the use of unsupervised models for language tasks. We can also touch on reinforcement learning methods and deep generative models. For Natural Language Processing (NLP) researchers, this is the first book that covers all the bases with regard to the most popular deep learning methods employed today [18-20]. This paper will help readers get a more complete picture of where things stand in this area. This paper is organized as follows: Section 2 introduces the definition of Natural Language Processing (NLP) and its approaches. Section 3 discusses popular NLP pipelines and fashions. Section four examines dimensionality discount strategies and section 5 specializes in modeling strategies that include type approaches. The evaluation of NLP models is given in section 6. Modern NLP applications and developments in unsupervised sentence representation learning are covered in section 7 and section 8 illustrates the recent trend towards trellising.

2. WHAT IS THE DEFINITION OF NATURAL LANGUAGE PROCESSING?

2.1. Natural Language Processing (NLP)

Linguistics, computer science, and artificial intelligence are all part of Natural Language Processing (NLP), an interdisciplinary discipline.

The development of computer systems that can efficiently handle and analyze large volumes of natural language input is its main emphasis within the field of computer-human interaction. Natural Language Processing (NLP) integrates statistical, machine learning, deep learning, and computational linguistics-based methodologies to model and analyze human language. Some of the most common challenges in NLP are speech recognition, NLU and NLG or natural language production, comprehension, and interpretation [2, 6]. Computer systems understand and manipulate human language with the help of these technologies. These days, natural language processing algorithms can sift through millions of web pages in a 2D space. Through its evolution, natural language processing (NLP) has gone from relying on symbolic processes to using statistical approaches and neural NLP techniques. Several state of the art natural language processing (NLP) applications use deep neural network architecture.

Generational innovations, improved processing capacity, and the availability of large corpora are all significant reasons. Human-to-human verbal exchange occurs through written and spoken language, forming the basis for Natural Language Processing (NLP). A subfield of computer science that allows the whole thing from email junk mail detection to predictive textual content [21]. Language and its evolution are studied through the lens of mathematical and computer modeling. NLP is machines can cause like people. Take a look at how computers and people communicate which is referred to as Natural Language Processing (NLP). The space between people and computer systems can be narrowed with the usage of natural language processing (NLP) [22].

2.2. Artificial Intelligence (AI)

Synthetic intelligence refers to the use of computers to replicate rational behavior with minimal human input. The aim of synthetic intelligence (AI) studies in PC science is to create PC programs with human-degree

intelligence [23, 24]. The goal of synthetic intelligence studies is to create laptop programs that can carry out responsibilities usually accomplished with the aid of human brains. A new and innovative synthetic intelligence era is being hailed as something on the way to modify the character of work [22].

Natural Language Processing (NLP) focuses on text and speech evaluation which will infer meaning from phrases. Recurrent Neural Networks (RNNs) which require deep knowledge of algorithms play a vital role in the processing of sequential inputs, including language, speech and time-collection records.

Deep getting to know is a subset of the device getting to know that can analyze unsupervised, unstructured or unlabeled data. On the other hand, deep learning can learn optimal features from available data without human intervention [25].

Natural language processing is the application of computer algorithms to the task of extracting meaning from unstructured spoken or written input by recognizing essential components of everyday language. Expertise in machine learning, computational linguistics and artificial intelligence is required for natural language processing [26].

Natural language processing (NLP) offers the following two techniques:

An approach based on rules where the computer follows the program's predetermined guidelines.

Approach incorporates ML-based supervised and unsupervised learning strategies. The difference between supervised and unsupervised learning is that the former involves human direction (annotation) to help computers acquire latent principles while the latter does not.

3. NLP PIPELINE

The Natural Language Processing (NLP) pipeline is a list of the steps needed to understand and analyze human words. A typical Natural Language Processing (NLP) workflow comprises the following phases (see Figure 1):

3.1. Data Acquisition

The gathering or creation of data is the initial stage of an NLP pipeline. The most common formats for such information are datasets, Hypertext Markup Language (HTML) pages, tweets, documents and logs [26].

3.2. Text Cleaning

The second step is to extract text and remove symbols, HTML markup, junk characters, etc. from the data, which may entail the actions listed below [26].

Many necessary words, including stop words, misspellings, slang, etc. can be found in most text and document data sets. Noise and extraneous characteristics in various algorithms including statistical and probabilistic learning algorithms can have a negative impact on system performance.

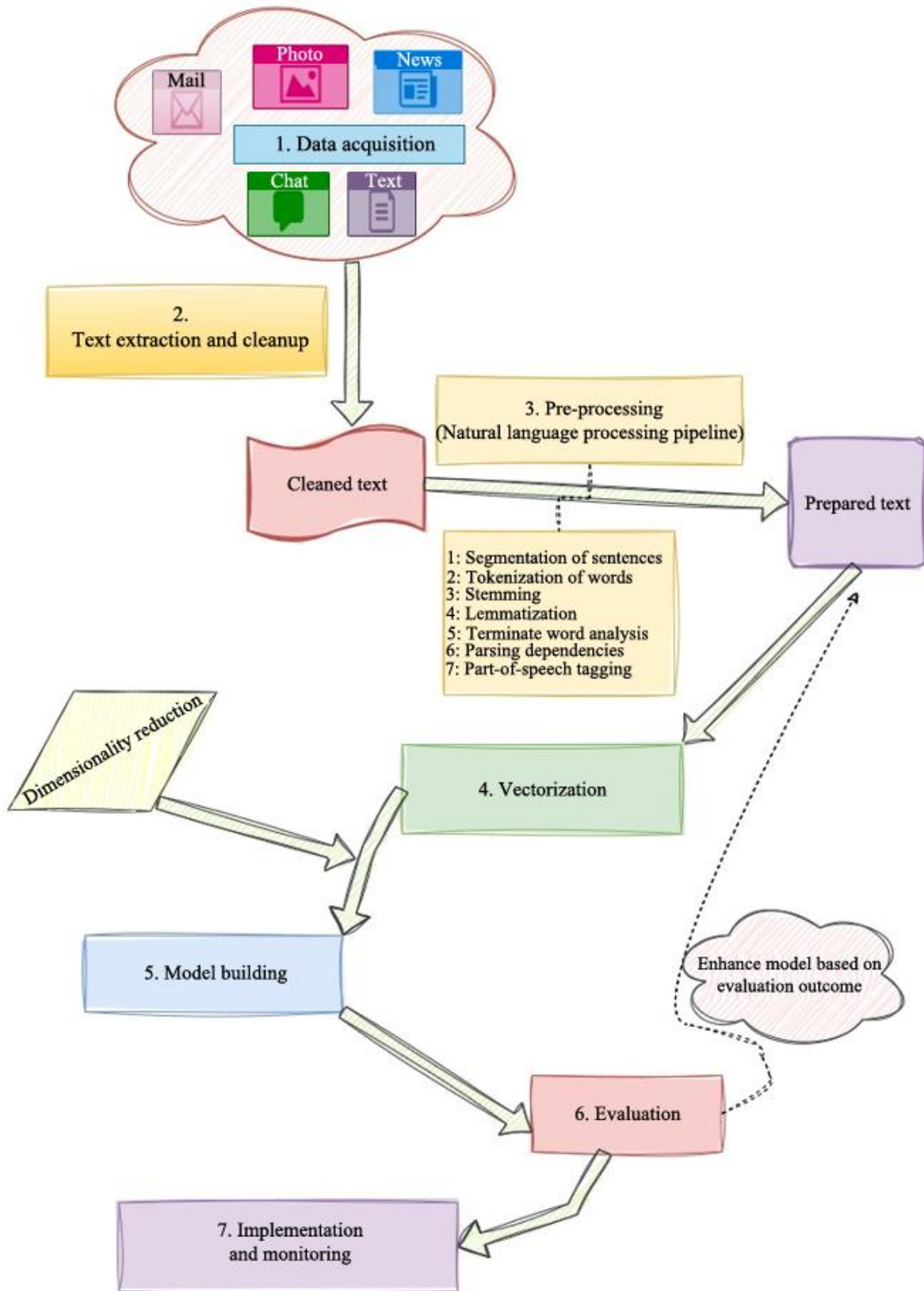


Figure 1. The Natural Language Processing (NLP) pipeline.

3.3. Pre-Processing

Pre-processing and feature extraction play crucial roles in text categorization applications. Propose techniques for cleaning text data sets that remove background noise so that useful functions may be performed.

Once all the text has been extracted and cleaned from the original data, it can undergo further processing. In this phase, convert this text data into sentences or words that can be used in subsequent steps of feature engineering. This step includes the following essential processes:

Step 1: Segmentation of Sentences (the detection of sentence boundaries)

Sentence segmentation is the initial phase in the pipeline for natural language processing. It breaks up the complete paragraph into separate sentences for clarity. A period typically indicates the detection of sentence boundaries.

Amman is the capital and largest city of Jordan. It is also the political, economic and cultural center of the country. Amman is the largest city in the Levant area and the fifth largest city in the Arab world.

After performing sentence segmentation, the following result is obtained:

Amman is the capital and largest city of Jordan.

It is also the political, economic, and cultural center of the country.

Amman is the largest city in the Levant area and the fifth largest city in the Arab world.

Step 2: Tokenization of words (splitting a sentence into separate tokens)

Word tokenization separates a phrase into individual words or tokens. This aids comprehension of the text's context. Amman, the capital and largest city of Jordan is tokenized as Amman, is , the , capital , and, largest, city, of , Jordan [27, 28].

Step 3: Stemming (reducing a word to its root)

Stemming facilitates text pre-processing. The model analyzes the elements of speech to determine the exact subject of the sentence.

The process of stemming converts words to their base or fundamental form. In other words, predicting the parts of speech for each token is beneficial. Intelligently, intelligence and intelligent are examples. These terms are derived from the root word intelligent. However, there is no word in English for "intelligent" [29].

Step 4: Lemmatization (token mapping)

Lemmatization eliminates inflectional endings and restores the canonical form or lemma of a word. Lemmatization is comparable to stemming except that the lemma is a real word. For instance, playing and plays are variants of the term play. Therefore, play is the lemma of these terms. In contrast to a stem (recall intelligent), play is a proper noun.

Step 5: Terminate word analysis

The following phase evaluates the significance of every word in a given sentence. Some English words such as is , and a, and the, appear more frequently than others. As they occur frequently, the NLP pipeline identifies them as stop terms. They are filtered out to concentrate on more significant words.

Step 6: Parsing dependencies

Next is dependency parsing which is primarily used to determine how all of the words in a sentence are connected. Construct a tree and designate a single word as its parent to determine the dependency. The root node of the sentence will be the primary verb.

Step 7: Part-of-Speech Tagging (POS labeling)

POS markers include verbs, adverbs, nouns, and adjectives that help indicate the grammatical meaning of words in a sentence [27].

Some other techniques could also be used.

3.3.1. Correction of Spelling Errors

The practice of correcting spelling errors is a pre-processing step that may be considered optional. Typographical errors are frequently faced in various forms of written communication, including texts and documents generally referred to as typos. These errors are particularly prevalent in datasets comprising social

media content such as those found on platforms like Twitter. Numerous algorithms, approaches and methodologies have been employed to tackle this issue in the field of Natural Language Processing (NLP) [30]. There are various strategies and methodologies that researchers can use such as hashing-based and context-sensitive spelling correction techniques as well as spelling correction utilizing Trie and POS labeling distance bigram.

3.3.2. Restrictive Verbs and Nouns

A large number of common words used in text and document classification such as a, about, above, across, after, afterwards, again, etc., lack sufficient meaning to be employed in classification algorithms. Removing these words from texts and documents is the most usual method of dealing with them.

3.3.3. Capitalization

Sentences formed from data points in text and documents contain varying degrees of capitalization. Since papers are made up of multiple sentences, different capitalization poses a significant challenge when trying to categorize them. It is customary practice to change all instances of mismatched capitalization to lowercase. This method maps all words in a text or document onto the same feature space. However, it creates serious issues when translating certain terms (such as UK (the United Kingdom) to UK (pronoun)). To accommodate these outliers, slang and abbreviation translators exist.

3.3.4. Abbreviations and Slang

Text oddities such as slang and abbreviations are also corrected during the pre-processing phase. Abbreviations, like SVM for Support Vector Machine (SVM) are condensed forms of words or phrases that primarily include the first letters of the original language. "Lost the plot" is an example of slang, a kind of phrase used in casual speech or writing with multiple meanings. Formalizing the language is a popular approach to dealing with these terms [30].

3.3.5. Noise Removal

Many of the text and document data sets have extraneous symbols, punctuation and other characters. Punctuation and other special characters can be problematic for categorization algorithms though essential to human comprehension.

3.4. Feature Engineering (Vectorization)

In this phase, extract features from the text input and feed them to the machine learning algorithm. In deep learning, feature extraction can be performed manually or with the assistance of a neural network. Both approaches have their own advantages and disadvantages. For example, if feature engineering is performed manually, one could easily determine how these features impact a model's performance [31]. However, in the case of deep learning-driven feature engineering, this information is unavailable as the neural network does not reveal the basis on which it selected a feature or its impact on the model's performance. In contrast, manual feature engineering necessitates task-specific domain knowledge which is unnecessary for a DL-driven approach [30] (see Figure 2).

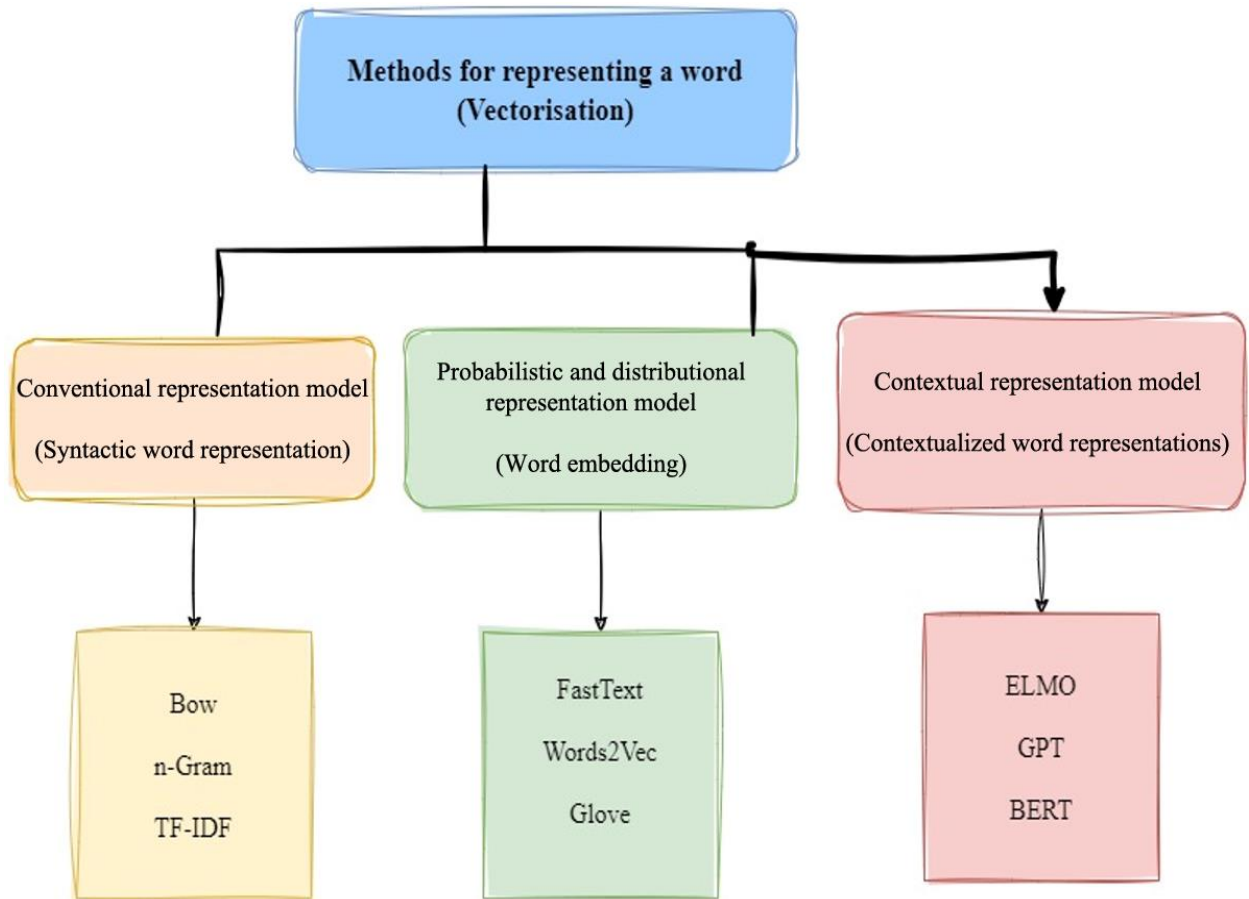


Figure 2. The methodology employed for the representation of a word.

3.4.1. Conventional Representation Model (Syntactic word representation)

3.4.1.1. Weighted Word Feature Extraction

Term Frequency (TF) represents a basic method of weighted word feature extraction where each word is allocated a value that reflects its frequency of occurrence throughout the entire corpus [32]. Approaches that enhance the outcomes of TF generally use Boolean or logarithmically scaled word frequency weighting [33].

In every word weighting method, each document is transformed into a vector (matching the document's length) that reflects the frequency of the words present in that document. This method presents limitations stemming from the tendency of certain commonly used words in a language to overshadow the overall representation (see Table 1) [34].

Table 1. Weighted word models: Advantages and limitations.

Model	Weighted words
Advantages	<ul style="list-style-type: none"> • It is straightforward to calculate the similarity between two documents using this method. • A fundamental metric for extracting the most descriptive keywords from a document. • Capable of handling unfamiliar words such as newly introduced terminology in languages.
Limitations	The current method fails to accurately represent the position, syntax and semantics of the text. Additionally, the presence of common terms like am and is negatively influences the quality of the findings.

3.4.2. Inverse Term Frequency Document Frequency

K. Sparck Jones introduced Inverse Document Frequency (IDF) as a method to be applied with term frequency to mitigate the influence of inherently common words within the corpus. The IDF assigns more significance to terms that exhibit either high or low frequency within the document [35].

This method is often known as Term Frequency-Inverse Document Frequency (TF-IDF). Equation 1 represents the mathematical representation of TF-IDF's calculation of a term's weight in a document.

$$W(d, t) = TF(d, t) * \log\left(\frac{N}{df(t)}\right) \quad (1)$$

For the purposes of this conversation, "t" stands for a single word and "d" for a document which is basically a group of words. "N" also stands for the count of the collection which is the total number of papers in the dataset.

A corpus is the whole set of papers that make up a collection.

The first term in Equation 1 makes it easier to remember things and the second term makes the word embedding more accurate [11]. Even though TF-IDF tries to solve the problem of document terms that are used too often, it still has some problems with how it describes things. To be more specific, TF-IDF can't figure out how related two words are. Each word is shown on its own as an index but as models have become more complicated in recent years, new techniques like word embedding that use ideas like word resemblance and part of speech labeling have been put forward (see Table 2).

Table 2. TF-IDF models: Advantages and limitations.

Model	TF-IDF
Advantages	<ul style="list-style-type: none"> • It is straightforward to calculate the similarity between two documents using this method. • A fundamental metric for identifying the most informative phrases within a document; • The impact of common words on the findings is mitigated by the use of inverse document frequency (IDF), rendering words like am and is inconsequential.
Limitations	The system fails to accurately represent the syntactic structure of the text, and it fails to accurately convey the intended meaning of the text.

3.4.3. Word Embedding

Although syntactic word representations are present, this does not indicate that the model effectively communicates the semantic meaning of the words. Conversely, bag-of-words models overlook the semantic meaning of words. The terms "aeroplan," "airplane," "plane," and "aircraft" are often used interchangeably. In the bag-of-words model, the vectors associated with these words are orthogonal. This issue presents a considerable challenge to understanding the model's sentences. Another limitation of the bag of words model is that it does not preserve the order of words in a phrase [11, 36]. The n-gram approach does not fully resolve this issue, necessitating the discovery of similarity for each word in a sentence. To address this, numerous researchers have employed word embedding techniques. Ruder and Søgaard [37] introduced the skip-gram and continuous bag-of-words (CBOW) models which use a straightforward, single-layer architecture based on the inner product of two-word vectors.

Word embedding is a feature-learning technique where each word or phrase in a vocabulary is represented as an N-dimensional vector of real numbers. Various methods have been proposed to convert unigrams into machine-readable input. This study focuses on three widely used and effective deep learning techniques: Word2Vec [38], GloVe [39], and FastText [40] are summarized in Table 3.

3.4.4. Word2Vec

Mikolov et al. [33] presented "word-to-vector" representation as an enhanced architecture for word embedding. Deep neural networks with two hidden layers, continuous bag-of-words (CBOW), and the skip-gram model are used by the Word2Vec method to turn each word into a high-dimensional vector. The skip-gram model by Patil et al. [35], Ruder and Søgaard [37] and Jang et al. [38] analyzes a corpus of words w and context c . The objective is to maximize the probability: $\arg \max_{w \in T} \sum_{c \in (w)} p(c | w; \theta)$ (2) where T denotes Text and θ is the parameter of $p(c | w; \theta)$.

$$\arg \max_{\theta} \prod_{w \in T} [\prod_{c \in c(w)} p(c/w; \theta)] \quad (2)$$

Table 3. Word2vec models: Advantages and limitations.

Model	Word2Vec
Advantages	The syntactic aspect of text analysis involves capturing the positional information of words within the text. On the other hand, the semantic aspect focuses on capturing the meaning conveyed by the words.
Limitations	The system exhibits limitations in capturing the semantic nuances of words within the given text as it fails to account for polysemy. Additionally, it is unable to accurately interpret terms that are not present in the corpus.

3.4.5. Global Vectors for Word Representation (GloVe)

Global Vectors (GloVe) [39] is another powerful word embedding technique that has been used for text classification. The strategy closely resembles the Word2Vec method in which each word is represented by a high-dimensional vector and trained based on the surrounding words in a massive corpus. GloVe offers additional pre-trained word vectorizations with 100, 200 and 300 dimensions that are trained on even larger corpora including Twitter content (see Table 4). The function of the objective is as follows (3):

$$(w_i - w_j, \tilde{w}_k) = \frac{p_{ik}}{p_{jk}} \quad (3)$$

Where w_i is the word vector for word i, and p_{ik} is the probability that word k will appear in the context of the word i.

Table 4. Glove models: Advantages and limitations.

Model	GloVe (pre-trained)- GloVe (trained)
Advantages	The system records the positional information of words in the text which pertains to syntax. Additionally, it captures the semantic meaning that the words intended to convey. The model was trained on a vast corpus. Enforcing sub-linear relationships in the vector space can be easily achieved. For instance, modification of word vectors. This approach has demonstrated superior performance compared to Word2vec. Weight should be reduced for word pairs that are highly frequent such as stop words like am, is, and so on. The lack of dominance will not impede the process of training.
Limitations	<ul style="list-style-type: none"> The text is unable to adequately grasp the multifaceted meaning of the word, therefore failing to portray its polysemy. The topic of interest pertains to the memory consumption associated with storage. The system is unable to recognise words that are not included in the corpus. The storage process consumes a significant amount of memory. A substantial corpus is required for effective learning. The model lacks the ability to recognise terms that are not present in the corpus. The model struggles to comprehend the multiple meanings of words within a given text.

3.4.6. FastText

Many other word embedding representations disregard the morphology of words by designating each word a unique vector [41] (see Table 5). The Facebook AI Research division came up with FastText, a brand-new word embedding technique as a creative solution to this issue [40].

An n-gram bundle of characters, w stands for each word. For instance, given the word "introduce" and n = 3, FastText will generate the following character tri-gram representation: <in, int, ntr, tro, rod, odu, duc, uce, ce >.

Please note that the sequence <int> that corresponds to the word here is distinct from the trigram "int" found in the word introduce. Suppose there is a dictionary of n-grams of size G and are given a word w for which a vector representation Z_g is associated with each n-gram g.

$$S(w, c) = \sum_{g \in g_w} z_g^T v_c \quad (4)$$

Where $g_w \in \{1, 2, \dots, G\}$

Facebook has released pre-trained word vectors for 294 languages that were trained on Wikipedia with FastText and 300 dimensions. The skip-gram model by Lazaridou et al. [42] was used by FastText with the preset parameters.

Recently, the "contextualized word representations" or "deep contextualized word representations" technique was introduced in which word vectors are dependent on the context of the word.

Table 5. Fasttext models: Advantages and limitations.

Model	Fasttext
Advantages	This approach is effective for words that are infrequently used as they possess unique character n-grams that are still present in other words. Additionally, it successfully addresses the issue of out-of-vocabulary terms by using n-gram analysis at the character level.
Limitations	The text fails to adequately reflect the polysemy of the term. Additionally, it is important to consider the memory usage for storage and the computational expense when comparing it to GloVe and Word2Vec.

3.4.7. Contextualized Word Representations (Transformer models)

Transformer models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pretrained Transformer) are potent deep learning models that have demonstrated outstanding performance in natural language processing (NLP) tasks. They are appropriate for tasks such as machine translation, text generation and language comprehension because they excel at recognizing contextual relationships in text [43].

3.4.8. Embeddings from Language Models (ELMo)

The ELMo by Peters [44] is a type of illustration that captures the semantic and syntactic information of words and sentences. These embeddings are generated through language fashions which are neural community-based fashions skilled in huge quantities of textual content information.

The ELMo representations are vectors that are obtained from a bidirectional LSTM (BiLSTM) that has been trained on a full-size corpus of text. The Elmo model efficiently tackles the assignment of expertise to the syntactic and semantic aspects of words as well as the linguistic contexts in which they may be applied. ELMo takes into consideration the entirety of a sentence when assigning a specific embedding to character phrases. The hired layout is bidirectional with phrase embeddings encouraged by using each of the preceding and following words in the sentence. The goal is to determine the best possibility of the language version in each ahead and backward instruction for a sequence of N tokens (t_1, t_2, \dots, t_N). The probability of the collection is calculated using a forward language version that estimates the probability of a token t_k given the history ($t_1, t_2, t_3, \dots, t_k$).

A backward language model is just like a forward language model in that it traverses the series. However, in the opposite order, predicting the previous token is based on the subsequent context (see Table 6).

The ahead and backward language versions are shown in Equations 5–7 as described by Peters in addition to the part of the expression that maximizes the logarithmic possibility in each direction [44].

$$P(t_1, t_2, \dots, t_N) = \prod_{k=1}^N P(t_k | t_1, t_2, \dots, t_{k-1}) \quad (5)$$

$$P(t_1, t_2, \dots, t_N) = \prod_{k=1}^N P(t_k | t_{k+1}, t_{k+2}, \dots, t_N) \quad (6)$$

$$\sum_{k=1}^N (\log p(t_k | t_1, t_2, \dots, t_{k-1}) + \log p(t_k | t_{k+1}, t_{k+2}, \dots, t_N)) \quad (7)$$

Table 6. Elmo models: Advantages and limitations.

Model	ELMo
Advantages	<p>ELMo is a language version that produces contextualized word embeddings in which the embeddings are sensitive to the surrounding context of a phrase resulting in various representations of the equal word in exclusive contexts. ELMo possesses the capability to comprehend subtleties and a couple of meanings in language, rendering it noticeably talented for jobs that necessitate comprehension of word semantics inside a given context.</p> <p>ELMo has the functionality to undergo pre-schooling on a vast corpus of textual facts, observed by way of best-tuning to cater to precise downstream wishes. The versatility of this era renders it tremendously versatile and treasured for a diverse array of natural language processing (NLP) activities, encompassing sentiment analysis named entity recognition and others.</p> <p>The embeddings produced by using ELMo are considered interpretable because of their era method involving a bi-directional LSTM. This characteristic enables comprehension for researchers and practitioners seeking insights into the selection-making system of the model.</p>
Limitations	<p>ELMo is a language version that produces contextualized word embeddings in which the embeddings are sensitive to the surrounding context of a phrase resulting in various representations of the equal word in exclusive contexts. ELMo possesses the capability to comprehend subtleties and a couple of meanings in language, rendering it noticeably talented for jobs that necessitate comprehension of word semantics inside a given context.</p> <p>ELMo has the functionality to undergo pre-schooling on a vast corpus of textual facts, observed by way of best-tuning to cater to precise downstream wishes. The versatility of this era renders it tremendously versatile and treasured for a diverse array of natural language processing (NLP) activities, encompassing sentiment analysis named entity recognition, and others.</p> <p>The embeddings produced using ELMo are considered interpretable because of their era method involving a bi-directional LSTM. This characteristic enables comprehension for researchers and practitioners seeking insights into the selection-making system of the model.</p>

3.4.9. Bidirectional Encoder Representations from Transformers (BERT)

BERT is a transformer-primarily-based model that has been pre-trained and may be tailor-made for unique NLP responsibilities. It learns contextual representations of words by analyzing both aspects of adjoining words. This bidirectional approach permits BERT to seize wealthy semantic data and perform well in a variety of tasks, such as sentiment analysis, named entity reputation and question answering [26, 45].

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained version educated on unlabeled text from BookCorpus and English Wikipedia. It can be pleasant-tuned for various NLP duties which include query answering, sentiment evaluation, textual content classification, sentence embedding, and ambiguity resolution [44]. BERT analyzes text bidirectional to achieve deeper language comprehension unlike earlier language fashions that processed textual content in one path and anticipated the following phrase for obligations like sentence era. Additionally, BERT presents contextual embeddings that adjust with the sentence in contrast to context-unfastened models along with word2vec and GloVe which generate a single vector for every word regardless of context. For example, within the sentences "he goes to the riverbank for a stroll" and "he is going to the bank to withdraw a few cash," BERT generates awesome vector representations for the word "financial institution" based totally on its context (see Table 7).

A lengthy text sequence must be divided into multiple short text sequences of 512 tokens due to the fact that BERT considers up to 512 tokens. This is a limitation of BERT as it is incapable of processing lengthy text sequences.

Table 7. Bert models: Advantages and limitations.

Model	BERT
Advantages	<p>The BERT model exhibits wider availability and pre-education in a couple of languages as compared to opportunity models. The utilization of non-English-based initiatives can prove to be superb in our paintings.</p> <p>When considering venture-unique fashions, BERT emerges as a beneficial choice. The BERT language model has been trained in the use of a greater big corpus, consequently facilitating its application to smaller and greater unique duties.</p> <p>The BERT version may be readily fine-tuned and directly deployed for various applications.</p> <p>BERT has a commendable degree of accuracy due to its normal updates.</p>
Limitations	<p>The BERT language model is characterized by using its high fee and multiplied computational necessities due to its large size.</p> <p>BERT has been particularly evolved to function as the input for many other structures. Additionally, BERT undergoes a technique known as high-quality tuning to optimize its overall performance for downstream duties which often showcase a high level of sensitivity and specificity.</p> <p>The length of the model is enormous because of the extensive corpus and the complicated schooling framework.</p> <p>The educational manner of BERT is characterized by its sluggishness due to its good-sized length and the giant quantity of weights that necessitate updating.</p>

3.4.10. Generative Pre-Training (GPT)

The utilization of GPT by Radford and Narasimhan [46] allows for the thorough exploration and utilization of word morphology within the application domain. The GPT model uses a unidirectional language model based on the transformer architecture for feature extraction whereas ELMo employs a bidirectional Long Short-Term Memory (BiLSTM) model (see Table 8).

The purpose of language modeling for a sequence of tokens (t1, t2,..., tN) is to maximize the likelihood as seen in Equation 8. The language model uses a multi-layer transformer decoder that includes a self-attention mechanism to expect the next word based totally on the preceding N-words [47]. The GPT model uses a multi-headed self-attention mechanism at the contextual tokens of the initial to ensure the right allocation of goal words. This is mixed with position-smart feed-forward layers as validated in Equations 9 to 11.

$$L_1(X) = \sum_i \log P(t_i / t_{i-N}, \dots, t_{i-1}; \theta) \quad (8)$$

$$h_0 = UW_e + W_p \quad (9)$$

$$h_1 = \text{transformer}_{\text{block}(h_{1-i}) \forall i} \in [1, n] \quad (10)$$

$$P(u) = \text{softmax}(h_n W_e^T) \quad (11)$$

Table 8. GPT models: Advantages and limitations.

Model	GPT models
Advantages	<p>The generative capacities of GPT models specifically the ones of substantial size which include GPT-3, enable them to produce textual content that carefully resembles human language, displaying a noteworthy capability for innovative textual content generation. This renders them high-quality for activities which include language technology, text augmentation, and chatbot development.</p> <p>Large-scale pre-education is not unusual guidance in education GPT models wherein those models are uncovered to tremendous volumes of textual data. This technique enables the acquisition of a various array of language styles and ideas as a result improving their applicability across many natural language processing (NLP) responsibilities.</p> <p>Transfer learning is a technique that involves fine-tuning GPT models using limited amounts of task-specific data enabling their adaptability to diverse applications.</p>
Limitations	<p>One hindrance to GPT is the absence of word-level embeddings. GPT is capable of producing textual content on the token degree such as sub word components or phrases; it no longer provides traditional word embeddings like Word2Vec or GloVe. The reliance on word-degree embeddings can provide barriers for certain programs.</p> <p>The absence of clean interpretability: GPT fashions are famed for their opaque characteristics, rendering comprehension in their selection-making processes hard. This lack of transparency may pose a substantial issue in contexts in which interpretability has the utmost significance.</p> <p>The length of the version: The deployment of large GPT models affords difficulties and necessitates big processing sources for each schooling and inference consequently restricting accessibility for a few users.</p>

This study focuses on the endeavor of generating a comprehensive evaluation of numerous natural language processing (NLP) models. A concise summary utilizing a visual representation in the form of a flowchart diagram is shown in Figure 3.

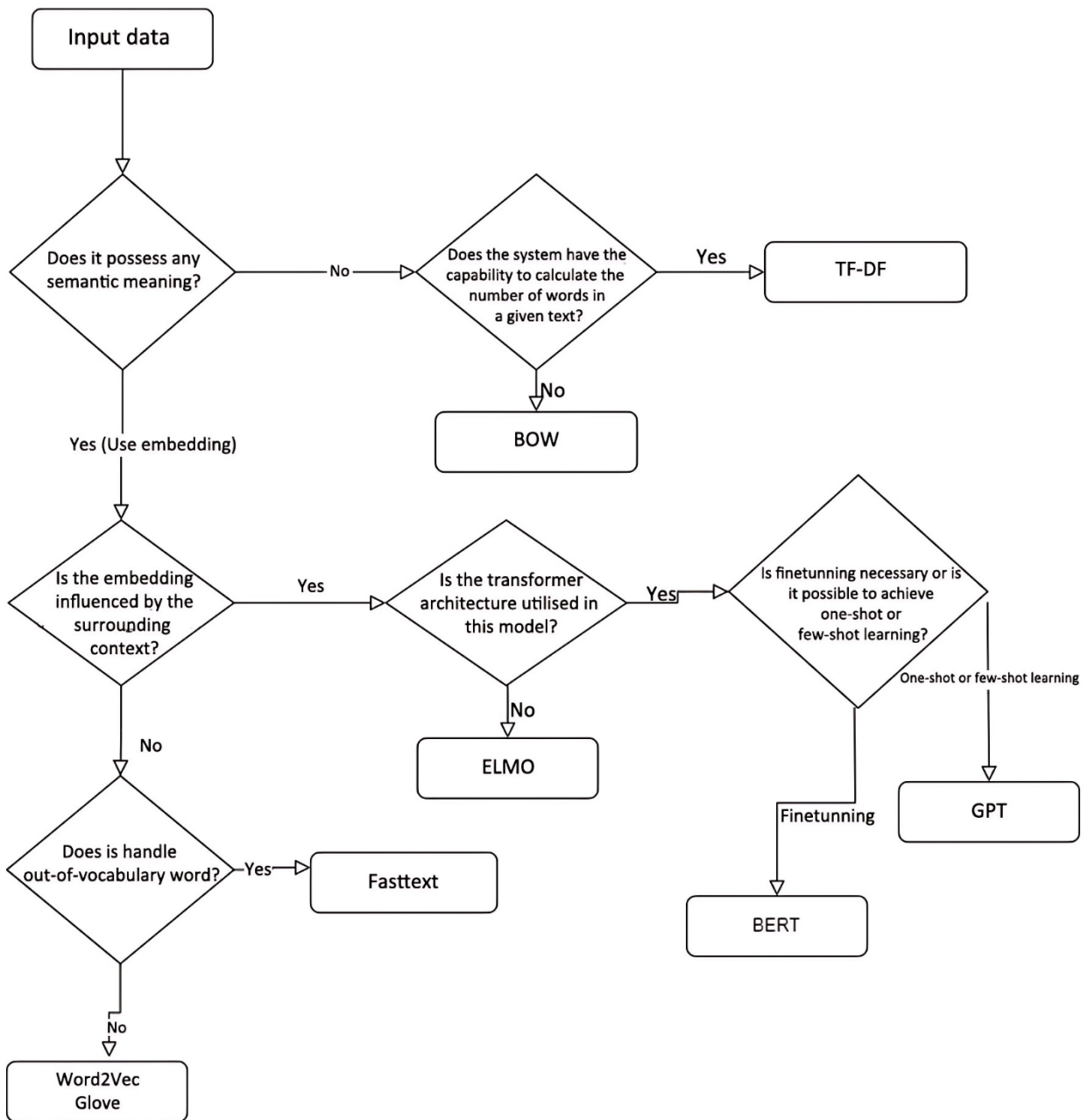


Figure 3. Various natural language processing (NLP) models.

4. DIMENSIONALITY REDUCTION

In term-based vector models, text sequences include many features. Consequently, the temporal complexity and memory use of these approaches are very high. A multitude of studies use dimensionality reduction to minimize the size of the feature space to tackle this problem. This section discusses existing dimensionality reduction methods.

4.1. Component Analysis

4.1.1. Principal Component Analysis (PCA)

Principal Issue Evaluation (PCA) is the most regular multivariate evaluation and dimensionality discount technique. It is a technique for identifying a subspace within which the facts kind of reside [48]. This includes

identifying new variables that are uncorrelated and maximizing variance to "keep as much variability as viable" [49].

It is utilized within the domains of records and data evaluation to extract capabilities and decrease dimensionality. This approach seeks to streamline the intricacy of excessive-dimensional facts while at the same time maintaining triumphant developments and styles. It also finds massive utility across various domains, encompassing gadgets gaining knowledge of, statistics visualization and picture processing.

PCA works by turning the initial statistics into a new coordinate system. The primary additives mix the original variables in a straight line along the new axes. The importance of these new variables is organized in the following order: the first important element is money owed for the best variance in the facts, the second for the following best, and so on. In this way, PCA aids in the reduction of information dimensionality while maintaining the greatest number of pertinent statistics [49].

Following are the procedures entailed in PCA:

- It is customary to standardize the data through the process of dividing the result obtained by subtracting the mean from the total and by the standard deviation of each variable to establish uniformity in the scale of the variables.
- Compute the covariance matrix: Utilize the standardized data to compute the covariance matrix. The covariance matrix illustrates the interrelationships among variables.
- Perform eigenvector and eigenvalue calculations: The eigenvectors and eigenvalues are computed for the covariance matrix. Eigenvectors indicate orientations with respect to the greatest variance whereas eigenvalues measure the proportion of variation that a certain eigenvector can explain.
- Sort eigenvectors: Arrange the eigenvectors so that the first eigenvector corresponds to the most significant principal component in descending order of their corresponding eigenvalues.
- Determine the number of principal components: The number of principal components to retain is determined by the percentage of total variance that they account for. One prevalent strategy is to preserve an adequate number of components to account for a specified proportion of the overall variance such as 95% of the variance.
- Data projection: Apply the selected principal components to the original data to generate a new dataset.

Principal Component Analysis (PCA) serves multiple functions such as diminishing the dimensionality of data, affording a reduced-dimensional representation of data and discerning patterns and interrelationships among variables. In machine learning, it is also employed to pre-process data prior to model training, a process that can result in enhanced model performance and accelerated training durations.

Consider a given data collection, denoted as $x(i)$ where i ranges from 1 to m . Each element $x(i)$ in the data set belongs to the n -dimensional real number space denoted as \mathbb{R}^n , for every i (where n and m are positive integers). The j th column of matrix X represents a vector denoted as x_j which comprises the observations pertaining to the j th variable. The expression representing the linear combination of x_j s can be denoted as:

$$\sum_{j=1}^m a_j x_j = Xa \quad (12)$$

The vector a represents a set of constants denoted as a_1, a_2, \dots, a_m . The variance of the linear combination can be expressed as follows:

$$\text{var}(Xa) = a^T S a \quad (13)$$

Let S be the sample covariance matrix. The objective is to identify the linear combination that exhibits the most variance. The objective is to maximize the function $a^T S a - \lambda(a^T a - 1)$ where λ represents the Lagrange multiplier.

PCA can be used as a pre-processing tool to reduce the dimension of a data set before executing a statistical analysis.

Supervised learning algorithm applied to it (x (i) s as inputs). PCA is also a valuable noise-reduction instrument.

Using the kernel approach, Kernel Principal Component Analysis (KPCA) is another dimensionality reduction technique that generalizes linear PCA to the non-linear case [49].

5. MODELING (CLASSIFICATION METHODS)

In this phase, an appropriate machine learning (ML) model will be selected based on the task and fed with the features from the previous step. This step's performance also depends on the preceding stages. A good model can produce poor results if the text input to feature engineering is not correctly processed. On the other hand, a simple model can yield the finest result if text processing and feature extraction are performed with care. The quantity of available data to train a model is also a significant factor [19, 20].

In this section, the research describes the classification algorithms currently used for text and documents. First, it describes some more traditional techniques such as logistic regression, Nave Bayes, and k-nearest neighbor, which are still widely employed in the scientific community. Support vector machines (SVMs) particularly kernel SVMs are also widely employed as classification methods. For categorizing documents, tree-based classification algorithms such as decision trees and random forests are efficient and accurate. Similarly, additionally, the research describes neural network-based algorithms, including deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN), and deep belief networks (DBN)[50, 51].

5.1. Logistic Regression

One of the earliest classification methods is logistic regression (LR) [52]. It is a linear classifier with $\theta^T x = 0$ decision boundary. Instead of predicting classes, LR predicts probabilities [51] (see Table 9).

Table 9. The pros and cons of logistic regression.

Model	Logistic regression
Advantages	<ul style="list-style-type: none"> • The proposed solution is characterized by its ease of implementation and its little demand for computer resources. • The scaling of input features (pre-processing) is not necessary. • No tweaking is required.
Drawback	<ul style="list-style-type: none"> • Non-linear issues cannot be solved using it. • The process of prediction necessitates the assumption that each data point is independent. • The act of making predictions is contingent upon utilizing a collection of independent variables to forecast events.

5.2. Naïve Bayes Classifier

The Naive Bayes text classification approach is extensively used for document categorization jobs. The Naive Bayes classifier is predicated on Bayes' theorem formulated by Thomas Bayes between 1701 and 1761 [52]. Recent studies have concentrated significantly on this method in information retrieval. This approach is a generative model, the predominant technique for text categorization [53, 54] (see Table 10).

Table 10. The pros and cons of naïve Bayes classifier.

Model	Naïve Bayes classifier
Advantages	The method has high efficacy when used to text data. It is characterized by its ease of implementation and notable speed advantages over alternative algorithms.
Drawback	One robust assumption on the distribution of data. When the feature space does not provide sufficient information for each potential value, the frequentist's estimation of a likelihood value is limited.

5.3. K-Nearest Neighbor

The k-nearest Neighbors algorithm (KNN) is a non-parametric classification technique. Decades ago, this method was used for text classification applications in a variety of research domains (see Table 11).

Table 11. The pros and cons of K-nearest neighbor.

Model	K-Nearest neighbor
Advantages	Text data sets can be effectively utilized. <ul style="list-style-type: none"> • A non-parametric approach is employed. • The analysis takes into account the text or document's local features to a greater extent. • The method inherently accommodates multi-class data sets.
Drawback	The computational cost associated with this model is prohibitively high. <ul style="list-style-type: none"> • The determination of the optimal value of k poses a challenge. • The identification of nearest neighbours in big search tasks is constrained.

5.3.1. Fundamental Concepts of KNN

Look at document (x), the KNN algorithm identifies the (k) nearest pals from the schooling set and calculates class scores based totally on the instructions of these buddies. The similarity between x and every neighboring file contributes to the score of the corresponding category [55]. If a couple of neighboring documents belong to the same elegance, their similarity rankings are summed to determine the overall rating for that class with recognize to x. . The algorithm then assigns (x) to the elegance with the highest cumulative score after rating all categories [55, 56]. The choice rule for KNN is as follows:

$$f(x) = \underset{j}{\operatorname{argmax}} S(x, C_j) = \sum_{d_i \in KNN} \operatorname{sim}(x, d_i) y(d_i, C_j) \quad (14)$$

5.4. Support Vector Machine (SVM)

Initially, SVM was developed for binary classification tasks. Nevertheless, a large number of researchers work on multi-class problems using this dominant technique [57-59]. Figure 4 depicts the linear and non-linear classifiers utilized for two-dimensional datasets (see Table 12).

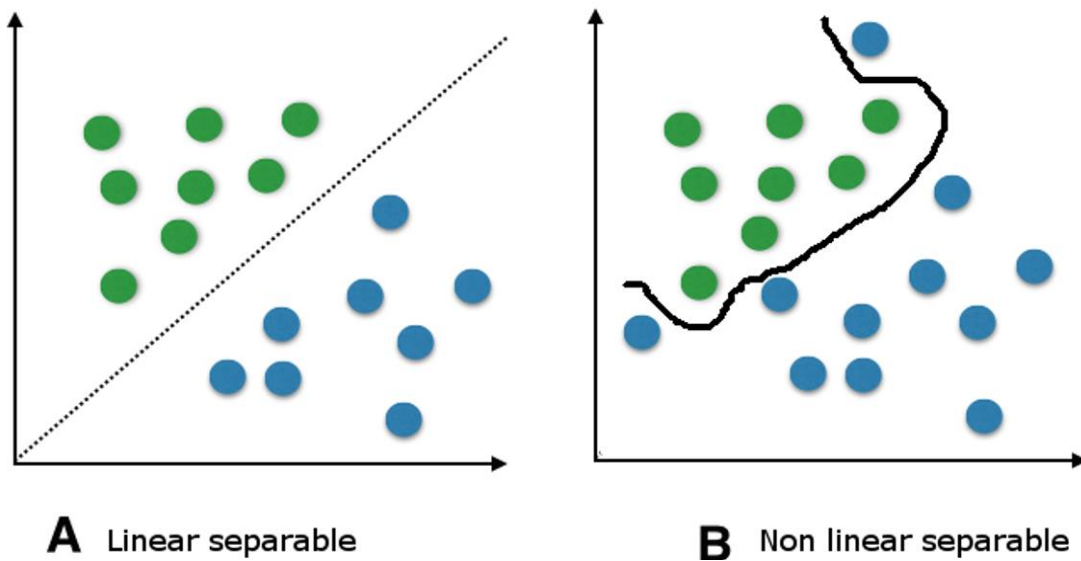


Figure 4. Depicts the linear and non-linear classifiers utilized for two-dimensional datasets.

Table 12. The pros and cons of support vector machine (SVM).

Model	Support vector machine (SVM)
Advantages	Support vector machines (SVM) have the ability to effectively represent non-linear decision boundaries. In situations where linear separation is feasible, SVM performs comparably to logistic regression. Additionally, SVM exhibits robustness against overfitting issues, particularly in the case of text datasets characterized by high-dimensional spaces.
Drawback	A lack of clarity in the results resulting from a large number of dimensions particularly for text data. <ul style="list-style-type: none"> • It is challenging to select an efficient kernel function depending on the kernel, overfitting or training issues may arise. • Memory complications.

5.5. Decision Tree

The decision tree is a traditional classification technique used in text and data mining [60]. Decision tree classifiers (DTCs) are used proficiently in several classification applications. The structure of this approach is a hierarchical breakdown of data space. The primary objective is to construct a tree using attributes for classified data points; nevertheless, the most difficult component of a decision tree is in identifying which attribute or feature should occupy the parent level and which should be positioned at the subordinate level for a training set with p positive and n negative examples.

$$H\left(\frac{p}{n+p}, \frac{n}{n+p}\right) = -\frac{p}{n+p} \log_2 \frac{p}{n+p} - \frac{n}{n+p} \log_2 \frac{n}{n+p} \quad (15)$$

Select attribute A which has k different values to partition the training set E into subsets {E1, E2, ..., Ek}.

The residual entropy denoted as EH is the measure of uncertainty that persists after attempting to determine attribute A which has k branches (i = 1, 2, ..., k).

$$EH(A) = \sum_{i=1}^K \frac{p_i+n_i}{p+n} H\left(\frac{p_i}{n_i+p_i}, \frac{n_i}{n_i+p_i}\right) \quad (16)$$

The measure of information gain (I) or reduction in entropy for this property is as follows:

$$A(I) = H\left(\frac{p}{n+p}, \frac{n}{n+p}\right) - EH(A) \quad (17)$$

Select the property that exhibits the most information gain to serve as the parent node (see Table 13).

Table 13. The pros and cons of decision tree.

Model	Decision tree
Advantages	The algorithm is capable of effectively managing qualitative (categorical) features. It demonstrates proficiency in handling decision boundaries that are parallel to the feature axis. Additionally, the decision tree algorithm is known for its high speed in both the learning and prediction processes.
Drawback	One of the challenges experienced in decision boundary construction is the presence of diagonal decision borders. <ul style="list-style-type: none"> • The phenomenon of overfitting can occur readily. • The model exhibits a high degree of sensitivity to minor fluctuations in the dataset. • Difficulties arise when attempting to make predictions on data that was not included in the training process.

5.6. Deep Learning Deep

Learning models have attained state of the art performance in several domains including a wide range of NLP applications [61]. Deep learning architectures for text and document classification include three fundamental parallel architectures. Each model is described in detail below [11].

5.6.1. Deep Neural Networks

Deep neural networks (DNNs) are dependent on more than one interconnected layer where every layer exclusively receives enter from the previous layer and sends output to the subsequent layer in the hidden component [62]. Figure five depicts the structure of a regular DNN. The input layer establishes a connection between the function space and the primary hidden layer employing techniques such as TF-IDF, phrase embedding or alternative function extraction methods. In the case of binary classification, the output layer is composed of a single node. Conversely, for multi-class classification, the output layer contains a number of nodes that corresponds to the number of classes present. Each DNN model is designed with varying numbers of nodes consistent with the layer depending on the specific utility. The DNN features as a trained discriminative model employing a trendy returned-propagation algorithm. It uses activation features like sigmoid (see Equation 18) and ReLU (see Equation 19). For the multi-magnificence category, the output layer employs a softmax characteristic as proven in Equation 20.

$$f(x) = \frac{1}{1+e^{-x}} \in (0,1) \quad (18)$$

$$f(x) = \max(0, x) \quad (19)$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1} e^{z_k}} \quad (20)$$

$$\forall j \in \{1, \dots, K\}$$

The objective is to acquire knowledge about the correlation between the input and target spaces, X and Y respectively by utilizing hidden layers based on a collection of sample pairs (x, y) where x belongs to X and y belongs to Y. The process of vectorizing the raw textual data results in a string which is the input in the context of text classification applications.

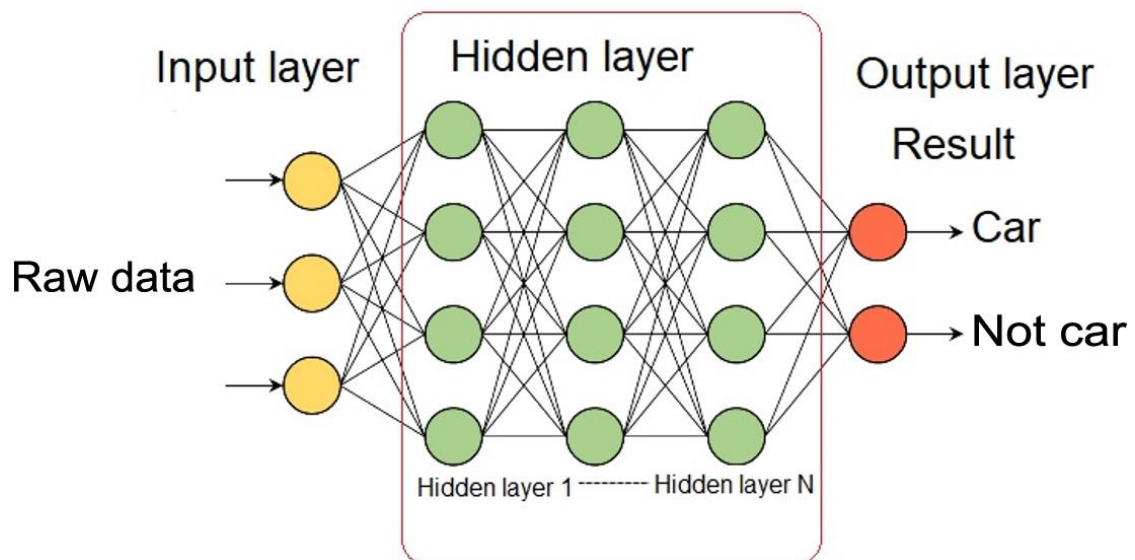


Figure 5. The structure of a typical DNN.

5.6.2. Recurrent Neural Network (RNN)

Another neural network structure that researchers use for text mining and type is a recurrent neural community (RNN) [63]. RNN assigns more weights to preceding data factors in a sequence. Therefore, this method is an effective method for classifying text, strings and sequential information. An RNN considers the records of preceding nodes in a noticeably sophisticated way permitting a more correct semantic evaluation of the shape of a data set [64, 65].

Long Short-Term Memory (LSTM) [66] and Gated Recurrent Unit (GRU) [67, 68] are variations of Recurrent Neural Networks (RNNs) that have been specially evolved to address a few shortcomings inherent in traditional RNNs such as the mission of vanishing or exploding gradients. This problem arises when the magnitudes of the gradients of the error feature turn out to be too small or big at some point in the backpropagation process as a result impeding the effective education of the neural community and the subsequent adjustment of the weights. LSTM and GRU models address the project at hand through the incorporation of gating mechanisms that alter the flow of information into and out of the hidden kingdom. These gates enable the community to gather the capacity to decide what data has to be retained and what facts must be overlooked. The Long Short-Term Memory (LSTM) model is equipped with three distinct gates, namely the input gate, output gate, and forget gate. In contrast, the Gated Recurrent Unit (GRU) model has two gates specifically the reset gate and the update gate.

According to Figure 6, RNNs typically employ LSTM or GRU for text classification with an input layer (word embedding), concealed layers and an output layer. This method can be expressed as follows:

$$x_t = f_{xt-1-u-J} \quad (21)$$

Where x_t is given by the state at time t , while u_t is the input at time t . Specifically, formulate Equation 21, which is parameterized by where W_{rec} represents the weight of the recurrent matrix, W_{in} represents the input weights, b represents the bias and an element-wise function. The architecture of an extended RNN is depicted in Figure 6. RNN is susceptible to the problems of vanishing gradients and exploding gradients when the error of the gradient descent algorithm is propagated back through the network. This is despite the benefits described above.

$$x_t = W_{rec} \sigma(x_{t-1}) + W_{in} u_t + b \quad (22)$$

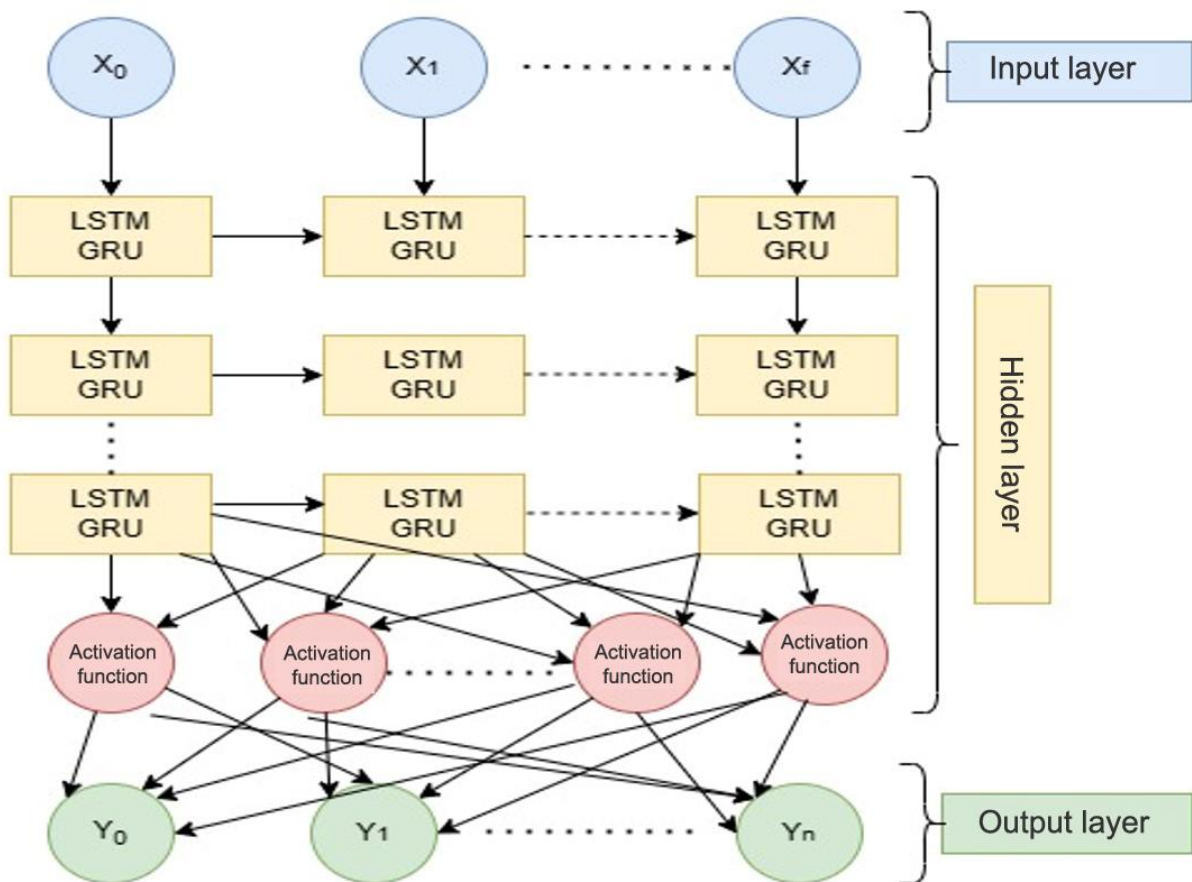


Figure 6. RNNs typically employ LSTM or GRU for text classification.

5.6.3. Long-Term Memory (LTM)

Numerous researchers have improved LSTM since its introduction by Schmidhuber [66]. LSTM is a specialized RNN that addresses these issues by preserving long-term dependency more efficiently than the standard RNN. LSTM is notably useful for overcoming the gradient-vanishing problem. LSTM employs multiple gates to regulate the amount of information allowed into each node state while both LSTM and RNN have a chain-like structure [68]. The following is a step-by-step explanation of an LSTM cell:

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \quad (23)$$

$$\hat{C}_t = \tanh(W_c[x_t, h_{t-1}] + b_c) \quad (24)$$

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \quad (25)$$

$$C_t = i_t * \hat{C}_t + f_t C_{t-1} \quad (26)$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \quad (27)$$

$$h_t = o_t \tanh(C_t) \quad (28)$$

Equation 23 represents the input gate. Equation 24 represents the value of the candidate memory cell. Equation 25 defines forget-gate activation. Equation 26 derives the new value of the memory cell. Equations 27 and 28 define the final output gate value. Each b in the preceding description represents a bias vector, each W represents a weight matrix, and x_t represents the input to the memory cell at time t . In addition, i , c , f , and o indicate input, cell memory, neglect, and output gates, respectively.

5.6.4. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are deep learning architectures often used for hierarchical file type [65, 69] originally designed for photograph processing. CNNs have also shown effectiveness in textual content classification. In a basic CNN for picture processing, an image sensor is convolved with a set of $d \times d$ kernels, forming convolutional layers known as characteristic maps. These layers can be stacked to provide multiple enter filters. CNNs rent aggregation techniques to reduce the output size between network layers to limit computational complexity. Pooling techniques, consisting of max pooling are normally used to condense the output while maintaining vital capabilities.

In max pooling, the largest detail within the pooling window is chosen. After pooling, the maps are flattened right into a single column to bypass the output from the stacked characteristic maps to the next layer. The very last layers of a CNN are usually fully linked. During lower back-propagation, the weights and function detector filters are up to date. A capability assignment in the usage of CNNs for text classification is the massive quantity of 'channels' (the dimensions of the feature space) which may be tons larger than in image class tasks. Snap shots generally have some channels (e.g ., RGB with 3 channels). Text class may additionally involve tens of lots of channels (e.g ., 50K), main to excessive dimensionality. Figure 7 illustrates the CNN structure for textual content class, inclusive of an input layer for word embeddings, 1D convolutional layers, 1D pooling layers, fully linked layers, and an output layer. Many researchers combine or concatenate well-known deep gaining knowledge of architectures to create novel and greater sturdy category techniques resulting in extra accurate models (see Table 14).

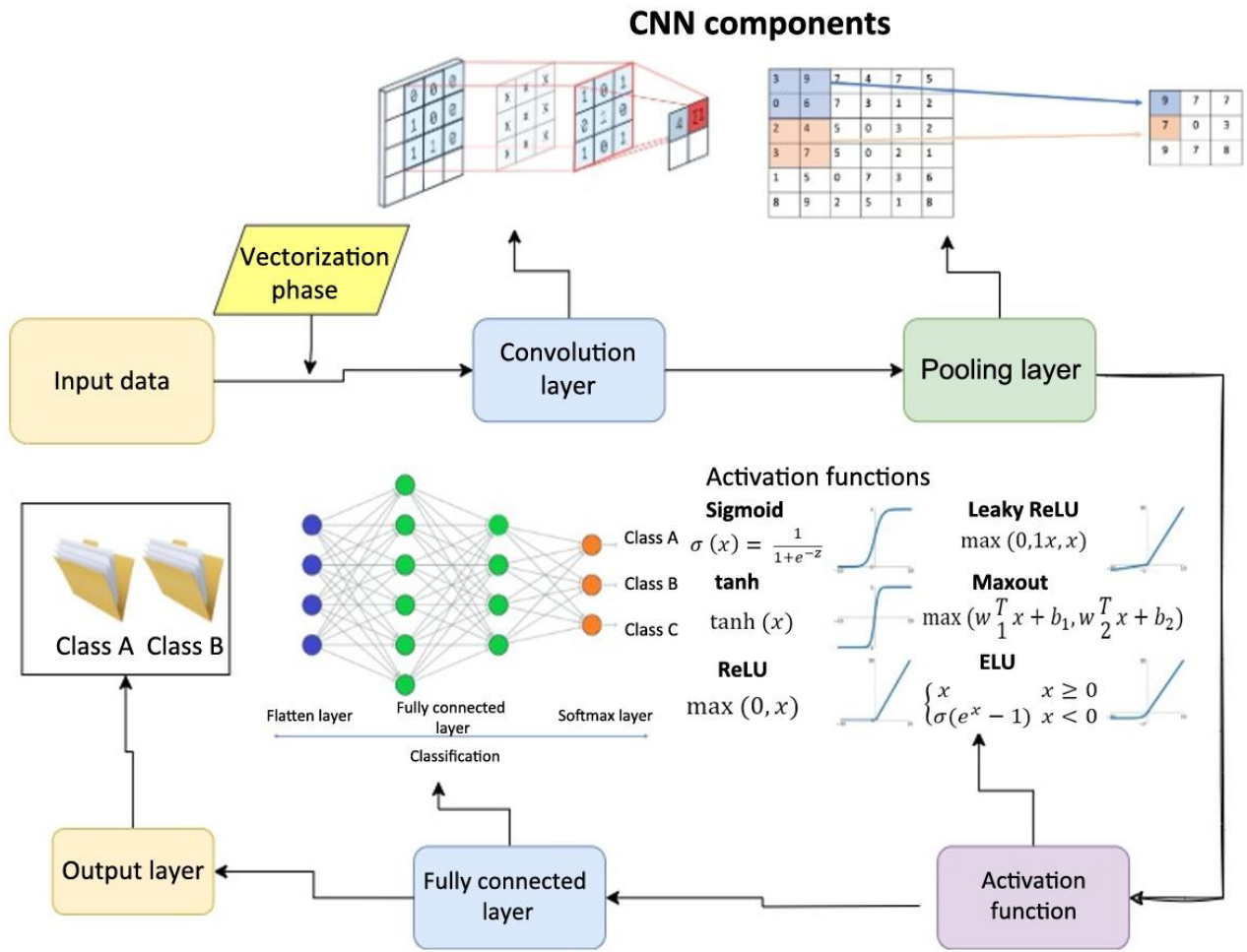


Figure 7. The CNN architecture for text classification.

Table 14. The pros and cons of deep learning.

Model	Deep learning
Advantages	<p>The proposed system exhibits flexibility in its design which effectively mitigates the necessity for feature engineering, a labor-intensive aspect of machine learning. Additionally, the architecture of the system is adaptable allowing for seamless integration with novel problem domains.</p> <p>The model exhibits proficiency in managing intricate input-output mappings and demonstrates a high degree of adaptability in accommodating online learning, facilitating the seamless retraining of the model with updated data.</p> <p>The system possesses the ability to engage in parallel processing, enabling it to execute multiple tasks simultaneously.</p>
Drawback	<p>It is imperative to have a substantial volume of data available for deep learning to achieve optimal performance. If the dataset consists of only a limited sample of text data, it is improbable that deep learning would surpass alternative methodologies.</p> <p>Training this model incurs a significant computational cost.</p> <p>The primary concern with deep learning is the lack of model interpretability as deep learning models often operate as black-box systems.</p> <p>The primary issue with this technique remains the identification of an efficient architecture and structure.</p>

6. EVALUATION

This phase determines the performance of the model on unknown data. This is highly dependent on the chosen metric for evaluation and the evaluation procedure itself. It also depends on the evaluation phases such as the modeling, deployment and production phases. The evaluation required for the first two phases is known as the

evaluation. Extrinsic evaluation can be performed in addition to intrinsic evaluation in the third phase to measure business impact using additional metrics. Listed below are a few evaluations conducted on the NLP model [70, 71].

In this research, it is ideal to evaluate algorithms using standardized and similar overall performance metrics. However, in practice, such measures are frequently only to be had for a limited quantity of strategies. One of the most important demanding situations in evaluating textual content classification techniques is the lack of standardized data acquisition protocols. Even if a standardized collection technique existed together with the Reuters news corpus, selecting one-of-a-kind schooling and taking a look at units may want to introduce inconsistencies in model performance. Additionally, evaluating the performance metrics utilized in distinct experiments can be difficult. Performance measures generally assess precise factors of a class challenge and as a result, they will now not usually deliver the same information. This phase explores numerous assessment metrics, performance measurements, and the evaluation of classifier overall performance. Since the underlying mechanisms of different assessment metrics can vary, it's important to sincerely apprehend what each metric represents and the form of statistics it ambitions to convey. Examples of such metrics consist of recall, precision, accuracy, f-degree, micro-average, and macro-common. These metrics are derived from a "confusion matrix" (see Table 15) that includes proper positives (TP), false positives (FP), fake negatives (FN) and true negatives (TN) [70]. The significance of those 4 elements may also vary depending on the category application. Accuracy is the share of accurate predictions relative to the entire variety of predictions (see Equation 29). Sensitivity, additionally called the true nice fee or bear in mind is the proportion of regarded positives which are efficiently anticipated (see Equation 30). Specificity is the proportion of effectively anticipated negatives (see Equation 31). Precision or positive predictive value is the ratio of successfully anticipated positives to all anticipated positives (see Equation 32).

$$accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (29)$$

$$sensitivity = \frac{(TP)}{(TP + FN)} \quad (30)$$

$$specificity = \frac{(TN)}{(TN + FP)} \quad (31)$$

$$precision = \frac{\sum_{l=1}^L TP_l}{\sum_{l=1}^L TP_l + FP_l} \quad (32)$$

$$recall = \frac{\sum_{l=1}^L TP_l}{\sum_{l=1}^L TP_l + FN_l} \quad (33)$$

$$F1 - Score = \frac{\sum_{l=1}^L 2TP_l}{\sum_{l=1}^L 2TP_l + TP_l + FN_l} \quad (34)$$

Comprehending the Confusion Matrix: The subsequent four concepts serve as fundamental terms that will aid in the determination of the desired metrics.

Table 15. Confusion matrix

Predicted values		Actual values	
		Positive	Negative
	Positive	True positives (TP)	True negatives (TN)
Negative	False positives (FP)	False negatives (FN)	

- True Positives (TP) occur when the actual value is classified as positive and the projected value is also classified as positive.
- True Negatives (TN) occur when the actual value is negative and the prediction value is similarly negative.
- False Positives (FP) occur when the actual value is negative but the prediction is positive.

- False negatives (FN) occur when the actual value is positive but the prediction is negative.

7. NLP APPLICATION

Many NLP applications have used various approaches to overcome the limitations of touching before feeling, seeing before imagining, and hearing before reacting. The development of text-to-speech and voice-to-text software has altered the basic nature of human reasoning. Threats to text and unhealthful contextual data can now be recognized using this innovative NLP application's spam detection feature. The sender is also responsible for deciding whether or not the message is genuine. If the recipient uses Gmail, you can check if an e-mail is already waiting for them in their inbox. Nowadays, spam text disappears after at least a month [72].

The methods for implementing an NLP application action and the practicality of each tool are detailed below. Speech, songs and word documentaries can all have their written words transformed into audio using natural language processing (NLP). Quickly assisting users is something that NLP excels at. Identify the main ideas, essential phrases and pertinent information in a work. The use of natural language processing aids readers in locating stress terms with ease [73]. The NLP software makes it simple to pinpoint the communication in question.

The goal is to identify and place into predetermined categories that appear in the unstructured text [74]. A text's unstructured content can be broken down into categories such as people, companies, locations, IDs, numbers and medical codes. Numbers, percentages and percentage points are particularly useful when working with massive datasets. Text extraction allows for the classification of unstructured material and the identification of pertinent information. Semantic analysis is a natural language processing application that analyzes the meaning of words, phrases, and sentences to help computers comprehend what they read (grammar, format and structure), establishing connections between words in a given setting [28]. Take the phrase "Sunday is awesome" as an example. The speaker is commenting on Sunday or the weekend. There are six stages to the natural language generation process. Content analysis. Here, the data is sorted so that only relevant pieces are included in the final product. Part of this procedure involves determining the central theme of the original text or document and tracing its interconnections.

Natural Language Processing (NLP) applications include machine translation, email spam detection, information extraction, summarization and question answering, among others. Next, some areas will be discussed where relevant work has been conducted.

7.1. Text Classification

Categorization systems receive a large volume of data such as official documents, market data, and newswires, and designate it into predefined categories or indices [75-77].

Some businesses have been employing categorization systems to route trouble tickets and complaint requests to the proper departments. E-mail spam filters are another use for text categorization. As the initial line of defense against unsolicited e-mail, spam filters are gaining importance. NLP technology has reduced the difficulty of extracting meaning from sequences of text by addressing the false-negative and false-positive issues that plague spam filters. The application of a filtering solution to an e-mail system employs a set of protocols to determine which incoming messages are spam and which are not. Various varieties of spam filters are available.

Content filters: Examine the message's content to determine whether or not it is spam.

Filter by Headers: Check if the header is fake. Block all emails coming from individuals on the blocked list.

Rule-Based Filters: It uses criteria defined by the user. For example, any email sent by a certain person or blocking. Any email with a certain word. Permission filters: Require the sender's permission before the recipient sends an email.

Challenge response filters would require the sender of an email to type in a code before granting permission to send it [75, 76].

7.2. Sentiment Analysis

Sentiment analysis also known as opinion mining is an activity within natural language processing (NLP) that seeks to identify and extract subjective information from source materials. This includes individual sentences or whole documents [78]. Recent advances in deep learning based dialogue systems. The system can automatically place attitudes into three categories which are neutral, negative, and positive. This valuable information can be gained from different sources such as customer reviews or social media sites. Sentiment classification is a task in which text data is assigned a sentiment class among various possible ones including positive and negative sentiments. For this purpose, different techniques have been used ranging from text preprocessing, feature extraction to machine learning models like Support Vector Machines (SVM), Naïve Bayes, Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), transformer-based methods such as GPT-3 and BERT [79]. This technology is widely used across many domains such as corporate decision-making processes, consumer feedback analysis, social media sentiment monitoring, political campaign planning, customer support systems and market research activities. These challenges include addressing sarcasm effectively, understanding context-based cues and adapting to multiple languages.

To mitigate these setbacks numerous tools and libraries have been developed by researchers and developers, for instance, NLTK, TextBlob, spaCy, VADER etc., including machine learning frameworks to facilitate this [80].

Opinion mining and sentiment analysis are critical components of understanding public mood in today's data-driven society. The field is constantly advancing through breakthroughs in natural language processing (NLP) models and techniques that enable corporations and organizations to make informed decisions based on public opinion and input.

7.3. Machine Translation

Most people in the world are now online which makes it hard to make sure that everyone can receive and use data. There is a language hurdle that makes it hard to receive information. There are a lot of languages, and each one has its own sentence structure and grammar rules. Machine translation usually means using a statistical tool like Google Translate to translate words and sentences from one language to another. The hard part about computer translation isn't changing words directly, but keeping the sense and language of lines [81].

The statistical machine learning system gathers as much similar data as it can between two languages. It then crunches the data to find the chance that something in language A is the same as something in language B. In September 2016, Google revealed a new machine translation system that is built on deep learning and artificial neural networks. Recently, many ideas have been put forward for letting machines automatically judge the quality of machine translations by comparing hypothesis translations with reference translations.

7.4. Spam Filtering

One of the primary challenges encountered by email users is the prevalence of spam mail which encompasses unsolicited material and data, including falsified content intended to disrupt individuals' lives. Additionally, certain emails may pose detrimental impacts on recipients. Currently, a significant proportion of individuals encompassing both those with formal education and those without are confronted with challenges pertaining to employment. Under such circumstances, individuals will receive electronic communications regarding promotional materials that are entirely fraudulent. Upon perceiving the aforementioned correspondence, individuals may develop an inclination towards engagement or contemplate initiating communication through electronic mail according to their specific areas of interest. Similar situations involve a larger number of people receiving these unsolicited emails. It is proposed to implement a system designed to effectively eliminate spam mail to mitigate this risk and safeguard individuals from the perils associated with unsolicited electronic communications. In this system, two filtering models are employed for spam mail filtration. Specifically, the two techniques that will be discussed are opinion

rank and an NLP-based n-grams model. Employing these two models will effectively classify and differentiate between spam emails and non-spam emails. The proposed system aims to improve data optimization through the use of a spam mail filtering mechanism and an email storage calculator [21].

7.5. Dialogue System

Dialogue systems also known as conversational systems have many applications such as personal assistants, voice control interfaces, and chatbots. They can communicate through speech, text, gestures and, visuals among others on both input and output channels [82, 83].

These systems provide users with an opportunity to engage in conversations for purposes of questioning, information acquisition, undertaking transactions (necessary ones), seeking help or support services, receiving recommendations or opinions from others or working towards achieving some goals. Some challenges associated with chatbots include their inability to interpret user questions which leads them to point the user towards the FAQ page instead of giving direct answers.

Moreover, chatbots have been unable to keep up with the whole conversation that they had with a particular user and may therefore answer questions given to them using irrelevant or too much information. These conversations are not particularly suitable for building dialogue production systems and over long periods of time do not perform well. Most chatbots are restricted in terms of domain specificity meaning that they can only operate within certain boundaries. The majority of these kinds of systems use primitive rule-based approaches. Dialogue systems have impressive performance in controlled conversational modes and question-answering sessions but lack flexibility and authenticity in portraying human-like conversations [82]. Nevertheless, it is important to note that despite these limits, the advantages of chatbots remain prominent. Such benefits include unrestricted availability prompt feedback and usage by individuals especially those who are afraid to ask questions publicly.

Most dialogue systems draw heavily on widely spoken languages such as Japanese, French German English etc. However, there are still some modern dialogue systems that need further expansion for other languages. The researchers face significant challenges when dealing with low-resource languages such as Slavic languages because they are highly inflectional. In Slavic languages, the large number of lexical items in a context is altered from their root forms due to contextual, morphological and grammatical changes. Additionally, numerals, adjectives, pronouns and nouns display phonetic and orthographic changes based on factors such as gender, number and grammatical case. Most Slavic languages have a rich noun inflection such as nominative, genitive, dative, accusative, locative, instrumental and vocative. Among these languages, the verb generally takes one of three tenses (future, present and past). Besides this quality 'aspect' may also be applied to the differentiation of verbs into two groups i.e. perfective indicating finished activities or imperfective referring to repeated/continuous actions. It must also be noted that gerunds and participles are more frequently used than clauses in English [81].

8. CHALLENGES AND ISSUES IN MODERN NATURAL LANGUAGE PROCESSING (NLP)

- Context understanding is limited. NLP models tend to have problems understanding context particularly for complex multi-turn dialogues.
- The possibility that models will capture and maintain biases that were already present in training data is another ongoing bias issue in natural language processing (NLP). Therefore, it is important to use every means at our disposal to promote bias reduction and equitable NLP systems.
- As far as Natural Language Processing (NLP) applications are concerned, there is a thin line between handling sensitive data for privacy concerns and operational efficiency.
- Multilingual and Cross-lingual Comprehension: Despite the significant strides made in developing natural language processing (NLP) specific to English; however, it still remains daunting to achieve comparable results for such languages that have scarce training data and promote cross-lingual comprehension.

- The question of interpretability and explainability looms large in the discussion about NLP. For this reason, more interpretable models need to be developed that could be used in transparent applications like healthcare or the legal sector.
- Natural language processing (NLP) low-resource languages is still a challenging and complex task.
- Pre-trained NLP models meant to understand domain-specific texts tend to perform badly making them require customized solutions instead.
- A growing research area focuses on reducing dependency on huge amounts of labeled data. To enhance the data efficiency of NLP models, few-shot learning together with data augmentation techniques is under investigation.

9. FUTURE DIRECTIONS IN NATURAL LANGUAGE PROCESSING (NLP)

- **Multimodal NLP:** Combining writing with other media types such as pictures, videos and sounds is a rising trend in NLP. Thus, multimodal NLP models are meant to comprehend or create materials across various media platforms other than writing only. As a result, it will be possible to know languages on a deeper level.
- **Conversational AI:** It is a milestone that makes chatbots and virtual assistants more natural and context-aware during the conversation. The ultimate goal of moving this technology forward is to lift up computer-reliant people working alongside computers and make virtual assistants smarter.
- **Interpretability and Transparency:** The demands for user-friendly NLP models have been growing exponentially across different sectors like banking, health and legal systems among others. This type of research seeks to explain how these models work so they can make better choices.
- **Enhancing the performance of natural language processing (NLP) models even when there is no or little training data is one of the most important objectives for future studies.** These include methods such as meta-learning and transfer learning aimed at helping models adapt to new tasks with minimal supervised data.
- **Better Pre-trained Models:** The search for larger or more superior pre-trained language models continues among researchers. Some of these architectures have emerged from ongoing efforts to enhance NLP competence for their applications towards more effective use across several domains.
- **Domain-unique Adaptation:** Instances like this are probably to come up. For instance, pre-skilled fashions can be specialized for areas consisting of healthcare or law making them more beneficial.
- **Zero-shot Translation:** One capability route involves the usage of NLP models that may translate between languages without the need for parallel training facts due to language barriers. As a result, this era can now reduce language boundaries extensively.
- **Understanding Emotions and Intents:** Innovations in detecting and generating feelings and intentions within textual content have a chief impact on sentiment evaluation applications and human-laptop interaction. Future looks at this area will be cognizant of growing NLP fashions.
- **Ethics in Artificial Intelligence:** Diversity is a key element to be taken into consideration even as conducting research and improving NLP, addressing worries about equity and bias in addition to ensuring that moral and accountable use of NLP era is guaranteed. The major attention for the future could be to make extra-ethical and fair NLP structures.
- **Useful Applications:** NLP is taking massive steps ahead into areas like weather technological know-how or clinical analysis (such as digital fitness records). There are many areas where NLP can make a big difference, and it continually finds new uses in the real world.

10. CONCLUSION

Technology is transforming the field of natural language processing which is changing how people interact with technology and analyze data. It has a variety of uses including customer service and content creation among

others. Recent improvements in transformer models as well as multimodal aspects have made it more powerful thus bringing new opportunities. However, despite its challenges, it appears that Natural Language Processing (NLP) can transform communication and understanding between humans and machines. As scholars continue to push the boundaries of knowledge, it is incontrovertible that Natural Language Processing (NLP) will be a significant force shaping technology and society.

This presentation will provide an in-depth analysis of natural language processing. From ancient beginnings to its present state, this paper traces the evolution of natural language processing. The way Natural Language Processing (NLP) is conducted now has changed; this method has shifted from symbolic or rule-based techniques to more advanced statistical methods and deep learning approaches. Some examples include BERT, GPT, ELMo models among others. The entry into this stage occurred with powerful deep learning approaches that enable computers to understand language meaningfully along the context. In relation to text and document categorization, the present study highlights pre-processing approaches, feature extraction and word embedding as essential components. It explains how these methods enhance the effectiveness of natural language processing (NLP) applications. Machine learning progress together with artificial intelligence development continuously proceeding within many other novel methods for better understanding human language augurs well for natural language processing's future.

The first objective provides an overview of important terminologies related to NLP which can be helpful for beginners who are just starting their career in NLP or related fields. Secondly, this paper will also examine the historical context along with the development and applications of NLP pipelines over time. The third objective was to explain NLP evaluation methods and metrics. Additionally, this paper will also present some of the most important works and projects in NLP as well as review the relevant work done in existing literature and its findings. The last two objectives can be a literature review for those who are already working in NLP or other related fields. They may also inspire readers to delve more into the topics discussed in this article. Even though there is a lot of literature about natural language processing (NLP) surveys focusing on one domain such as usage of deep-learning techniques in NLP, techniques used for e-mail spam filtering, management research, intrusion detection, Gujarati language, etc., work on regional languages is still scanty and could be the subject of future studies.

Funding: This study received no specific financial support.

Institutional Review Board Statement: Not applicable.

Transparency: The authors state that the manuscript is honest, truthful, and transparent, that no key aspects of the investigation have been omitted, and that any differences from the study as planned have been clarified. This study followed all writing ethics.

Competing Interests: The authors declare that they have no competing interests.

Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

REFERENCES

- [1] I. Lauriola, A. Lavelli, and F. Aioli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443-456, 2022. <https://doi.org/10.1016/j.neucom.2021.05.103>
- [2] H. Xu and K. Roberts, "Natural language processing. In intelligent systems in medicine and health: The role of AI." Cham: Springer International Publishing, 2022, pp. 213-234.
- [3] K. Chowdhary and K. R. Chowdhary, "Natural language processing," *Fundamentals of Artificial Intelligence*, pp. 603-649, 2020. <https://doi.org/10.5715/jnlp.27.963>
- [4] Z. Kaddari, Y. Mellah, J. Berrich, M. G. Belkasm, and T. Bouchentouf, "Natural language processing: Challenges and future directions. In international conference on artificial intelligence & industrial applications." Cham: Springer International Publishing, 2020, pp. 236-246.

- [5] H. A. Uymaz and S. K. Metin, "Vector based sentiment and emotion analysis from text: A survey," *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104922, 2022. <https://doi.org/10.1016/j.engappai.2022.104922>
- [6] M. H. Ullah, B. Jahan, A. Hoque, S. C. Paul, and M. S. Uddin, "A comparative study on evolution of natural language processing and natural language understanding," *Lecture Notes in Networks and Systems*, vol. 385, pp. 11–21, 2022. https://doi.org/10.1007/978-981-16-8987-1_2
- [7] S. Quarteroni, "Natural language processing for the web," in *Web Engineering: 12th International Conference, ICWE 2012, Berlin, Germany, July 23-27, 2012. Proceedings 12 (pp. 508-509)*. Springer Berlin Heidelberg, 2012.
- [8] M. M. Taye, R. Abulail, and M. Al-Oudat, "An ontology learning framework for unstructured arabic text," in *ISAS 2023 - 7th International Symposium on Innovative Approaches in Smart Technologies, Proceedings, 2023*, 2023.
- [9] A. Farzindar and D. Inkpen, "Natural language processing for social media, second edition," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 2, pp. 1–197, 2018.
- [10] A. Farzindar, D. Inkpen, and G. Hirst, *Natural language processing for social media*. San Rafael: Morgan & Claypool, 2015.
- [11] Y. Li and T. Yang, "Word embedding for understanding natural language: A survey," *Guide to Big Data Applications*, vol. 26, pp. 83-104, 2018.
- [12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
- [13] M. M. Lopez and J. Kalita, "Deep learning applied to NLP," Retrieved: <http://arxiv.org/abs/1703.03091>. [Accessed 2023].
- [14] F. Ahmad *et al.*, "A deep learning architecture for psychometric natural language processing," *ACM Transactions on Information Systems*, vol. 38, no. 1, pp. 1-29, 2020.
- [15] E. H. Houssein, R. E. Mohamed, and A. A. Ali, "Machine learning techniques for biomedical natural language processing: A comprehensive review," *IEEE Access*, vol. 9, pp. 140628-140653, 2021. <https://doi.org/10.1109/access.2021.3119621>
- [16] M. M. Taye, "Framework for enhanced ontology alignment using BERT-based," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 10, pp. 2853–2861, 2023.
- [17] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," Retrieved: <https://aclanthology.org/C18-1182>. 2018.
- [18] Y. Goldberg, "A primer on neural network models for natural language processing," *arXiv preprint arXiv:1703.00456*, 2017.
- [19] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017.
- [20] Y. Goldberg, Y. Liu, and M. Zhang, "Neural network methods for natural language processing," *Computational Linguistics*, vol. 44, no. 1, pp. 193–195, 2018.
- [21] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, vol. 29, pp. 63-92, 2008. <https://doi.org/10.1007/s10462-009-9109-6>
- [22] M. Alsaleh and A. Alarifi, "Analysis of web spam for non-english content: Toward more effective language-based classifiers," *PloS One*, vol. 11, no. 11, p. e0164383, 2016. <https://doi.org/10.1371/journal.pone.0164383>
- [23] J. Jeong, "Artificial intelligence in medicine," *Hanyang Medical Reviews*, vol. 37, no. 2, p. 47, 2017. <https://doi.org/10.7599/HMR.2017.37.2.47>
- [24] M. M. Taye, "Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions," *Computers*, vol. 12, no. 5, p. 91, 2023. <https://doi.org/10.3390/computers12050091>
- [25] O. Ajao, D. Bhowmik, and S. Zargari, "Fake news identification on Twitter with hybrid CNN and RNN models," in *ACM International Conference Proceeding Series*, pp. 226–230, 2018.
- [26] M. Pota, M. Ventura, R. Catelli, and M. Esposito, "An effective BERT-based pipeline for Twitter sentiment analysis: A case study in Italian," *Sensors*, vol. 21, no. 1, p. 133, 2020. <https://doi.org/10.3390/s21010133>

- [27] N. Habash and O. Rambow, "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop," in *ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 573–580, 2005, 2005.
- [28] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3713–3744, 2023. <https://doi.org/10.1007/s11042-022-13428-4>
- [29] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 275-282), 2002.
- [30] M. B. Akçay and K. Oğuz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers," *Speech Communication*, vol. 116, pp. 56-76, 2020. <https://doi.org/10.1016/j.specom.2019.12.001>
- [31] A. Abbasi, S. France, Z. Zhang, and H. Chen, "Selecting attributes for sentiment classification using feature relation networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 447-462, 2010. <https://doi.org/10.1109/tkde.2010.110>
- [32] F. P. Shah and V. Patel, "A review on feature selection and feature extraction for text classification," in *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, pp. 2264–2268, 2016.
- [33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Proceedings of NIPS*, vol. 2013, pp. 3111–3119, 2013.
- [34] R. Othman, R. Faiz, Y. Abdelsadek, K. Chelghoum, and I. Kacem, "Deep hybrid neural networks with improved weighted word embeddings for sentiment analysis " in *Advances in Intelligent Data Analysis XIX: 19th International Symposium on Intelligent Data Analysis, IDA 2021, Porto, Portugal, April 26–28, 2021, Proceedings 19* (pp. 50-62). Springer International Publishing, 2021.
- [35] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A survey of text representation and embedding techniques in nlp," *IEEE Access*, vol. 11, pp. 36120–36146, 2023. <https://doi.org/10.1109/access.2023.3266377>
- [36] X. Rong, "word2vec parameter learning explained," Retrieved: <http://bit.ly/wevi-online>. 2016.
- [37] S. Ruder and A. Søgaard, "A survey of cross-lingual word embedding models," *Journal of Artificial Intelligence Research*, vol. 65, pp. 569–631, 2019.
- [38] B. Jang, I. Kim, and J. W. Kim, "Word2vec convolutional neural networks for classification of news articles and tweets," *PloS One*, vol. 14, no. 8, p. e0220976, 2019. <https://doi.org/10.1371/journal.pone.0220976>
- [39] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1532–1543, 2014.
- [40] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *The Association for Computational Linguistics*, vol. 2, pp. 427–431, 2023.
- [41] A. Balamurali and B. Ananthanarayanan, "Develop a neural model to score bigram of words using bag-of-words model for sentiment analysis in neural networks for natural language processing," in *Proceedings of the International Conference on Computer Science and Information Technology*, pp. 275–279, 2015.
- [42] A. Lazaridou, N. T. Pham, and M. Baroni, "Combining language and vision with a multimodal skip-gram model," in *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pp. 153–163, 2015.
- [43] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 2978–2988, 2020.

- [44] M. E. Peters, "Deep contextualized word representations," in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1*, pp. 2227–2237, 2018.
- [45] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1*, pp. 4171–4186, 2019.
- [46] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training, OpenAI," Retrieved: <https://openai.com/research/language-unsupervised>. [Accessed: Dec. 11, 2024]. 2018.
- [47] G. Yenduri, "Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions," Retrieved: <https://arxiv.org/abs/2305.10435v2>. 2023.
- [48] S. Karamzadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman, "An overview of principal component analysis," *Journal of Signal and Information Processing*, vol. 4, no. 3, pp. 173-175, 2013.
- [49] F. L. Gewers *et al.*, "Principal component analysis: A natural approach to data exploration," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1-34, 2021.
- [50] S. Merity, N. S. Keskar, and R. Socher, "An analysis of neural language modeling at multiple scales," Retrieved: <http://arxiv.org/abs/1803.08240>. 2023.
- [51] Y. Zhou, "Natural language processing with improved deep learning neural networks," *Scientific Programming*, vol. 2022, no. 1, p. 6028693, 2022. <https://doi.org/10.1155/2022/6028693>
- [52] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons. <https://doi.org/10.1002/9781118548387>, 2013.
- [53] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: Logistic regression," *Nat Methods*, vol. 13, no. 7, pp. 541–542, 2016.
- [54] V. B. Vikramkumar and Trilochan, "Bayes and naive bayes classifier," Retrieved: <https://arxiv.org/abs/1404.0933v1>. [Accessed 2014.
- [55] F. J. Yang, "An implementation of naive bayes classifier," in *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, pp. 301–306, 2018.
- [56] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naïve Bayes algorithm," *Knowledge-Based Systems*, vol. 192, p. 105361, 2020. <https://doi.org/10.1016/j.knosys.2019.105361>
- [57] J. Laaksonen and E. Oja, "Classification with learning k-nearest neighbors," *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 3, pp. 1480–1483, 1996. <https://doi.org/10.1109/ICNN.1996.549118>
- [58] Y. Zhang, G. Cao, B. Wang, and X. Li, "A novel ensemble method for k-nearest neighbor," *Pattern Recognition*, vol. 85, pp. 13–25, 2019. <https://doi.org/10.1016/j.patcog.2018.08.003>
- [59] N. Cristianini and E. Ricci, "Support vector machines," *Encyclopedia of Algorithms*, pp. 928–932, 2008. https://doi.org/10.1007/978-0-387-30162-4_415
- [60] M. Awad and R. Khanna, "Support vector machines for classification," *Efficient Learning Machines*, pp. 39–66, 2015. https://doi.org/10.1007/978-1-4302-5990-9_3
- [61] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2049, pp. 249–257, 2001. https://doi.org/10.1007/3-540-44673-7_12
- [62] M. Fratello and R. Tagliaferri, "Decision trees and random forests," *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1-3, pp. 374–383, 2018. <https://doi.org/10.1016/B978-0-12-809633-8.20337-3>
- [63] P. Bhattacharyya, "Natural language processing meets deep learning," *Language Studies in India*, pp. 199–214, 2023. https://doi.org/10.1007/978-981-19-5276-0_12

- [64] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604-624, 2020. <https://doi.org/10.1109/tnnls.2020.2979670>
- [65] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, 1997. <https://doi.org/10.1109/78.650093>
- [66] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85-117, 2015. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [67] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, 2023. <https://doi.org/10.3390/computation11030052>
- [68] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computer*, vol. 9, no. 8, pp. 1735-1780, 1997. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [69] A. F. M. Agarap, "A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data," in *ACM International Conference Proceeding Series*, pp. 26-30, 2017.
- [70] A. Lawi, H. Mesra, and S. Amir, "Implementation of long short-term memory and gated recurrent units on grouped time-series data to predict stock prices accurately," *Journal of Big Data*, vol. 9, no. 1, p. 89, 2022. <https://doi.org/10.1186/s40537-022-00597-0>
- [71] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1746-1751, 2014.
- [72] H. Yang, L. Luo, L. P. Chueng, D. Ling, and F. Y. L. Chin, "Deep learning and its applications to natural language processing," *Cognitive Computation Trends*, pp. 89-109, 2019. https://doi.org/10.1007/978-3-030-06073-2_4
- [73] Z. Pauzi and A. Capiluppi, "Applications of natural language processing in software traceability: A systematic mapping study," *Journal of Systems and Software*, vol. 198, p. 111616, 2023. <https://doi.org/10.2139/ssrn.4170366>
- [74] T. P. Nagarhalli, S. Mhatre, S. Patil, and P. Patil, "The review of natural language processing applications with emphasis on machine learning implementations," presented at the International Conference on Electronics and Renewable Systems (ICEARS), pp. 1353-1358, 2022.
- [75] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019. <https://doi.org/10.3390/info10040150>
- [76] P. Pham, L. T. Nguyen, W. Pedrycz, and B. Vo, "Deep learning, graph-based text representation and classification: A survey, perspectives and challenges," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 4893-4927, 2023. <https://doi.org/10.1007/s10462-022-10265-7>
- [77] H. Li and Z. Li, "[Retracted] Text classification based on machine learning and natural language processing algorithms," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 3915491, 2022. <https://doi.org/10.1155/2022/3915491>
- [78] B. Babych and A. Hartley, "Improving machine translation quality with automatic named entity recognition," in *Proceedings of the 7th International EAMT Workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools, Resource and Tools for Building MT at EACL 2003*, 2003.
- [79] J. Ni, T. Young, V. Pandelea, F. Xue, and E. Cambria, "Recent advances in deep learning based dialogue systems: A systematic survey," *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3055-3155, 2023. <https://doi.org/10.1007/s10462-022-10248-8>
- [80] A. Ohashi and R. Higashinaka, "Adaptive natural language generation for task-oriented dialogue via reinforcement learning," Retrieved: <https://aclanthology.org/2022.coling-1.19>. [Accessed Aug. 25, 2024]. 2022.
- [81] E. Razumovskaia, G. Glavaš, O. Majewska, E. M. Ponti, and I. Vulić, "Natural language processing for multilingual task-oriented dialogue," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 44-50, 2022.

- [82] A. Abdellatif, K. Badran, D. E. Costa, and E. Shihab, "A comparison of natural language understanding platforms for chatbots in software engineering," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 3087-3102, 2021. <https://doi.org/10.1109/tse.2021.3078384>
- [83] A. Miklosik, N. Evans, and A. M. A. Qureshi, "The use of chatbots in digital business transformation: A systematic literature review," *Ieee Access*, vol. 9, pp. 106530-106539, 2021. <https://doi.org/10.1109/access.2021.3100885>

Views and opinions expressed in this article are the views and opinions of the author(s), Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.