IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 April 2024, accepted 23 May 2024, date of publication 28 May 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406426

TOPICAL REVIEW

A Review of State of the Art Deep Learning Models for Ontology Construction

TSITSI ZENGEYA AND JEAN VINCENT FONOU-DOMBEU
School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg 3209, South Africa
Corresponding author: Tsitsi Zengeya (222131004@stu.ukzn.ac.za)

ABSTRACT Researchers are working towards automation of ontology construction to manage the ever-growing data on the web. Currently, there is a shift from the use of machine learning techniques towards exploration of deep learning models for ontology construction. Deep learning model are capable of extracting terms, entities, relations, and classifications, and perform axiom learning from the underutilized richness of web-based knowledge. There has been remarkable progress in automatic ontology creation using deep learning models since they can perform word embedding, long-term dependency acquisition, concept extraction from large corpora, and inference of abstracted relationships based on broad corpora. Despite their emerging importance, deep learning models remain underutilized in ontology construction, and there is no comprehensive review of their application in ontology learning. This paper presents a comprehensive review of existing deep-learning models for the construction of ontologies, the strength and the weaknesses presented by the deep learning models for ontology learning as well as promising directions to achieve a more robust deep learning models. The Deep Learning models reviewed include Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long-Short Term Memory (LSTMs), and Gated Recurrent Unit (GRU) as well as their ensembles. While these traditional deep learning models have achieved great success, one of their limitations is that they struggle to understand the meaning and order of data in sequences. CNNs and RNN-based models such as LSTMs and GRUs can be computationally expensive due to their large number of parameters or complex gating mechanisms. Furthermore, RNN models suffer from vanishing gradients, making it difficult to learn long-term relationships in sequences. Additionally, RNN-based models process information sequentially, limiting their ability to take advantage of powerful parallel computing hardware, slowing down training and inference, especially for long sequences. Consequently, there has been a shift towards Generative Pre-Trained (GPT) models and Bidirectional Encoder Representations from Transformers (BERT) models. This paper also reviewed the GPT-3, GPT-4, and the BERT models for extracting terms, entities, relations, and classifications. While GPT models excel in contextual understanding and flexibility, they fall short when handling domain-specific terminology and disambiguating complex relationships. Fine-tuning and domain-specific training data could minimize these shortcomings, and further enhance the performance of GPT in term and relation extraction tasks. On the other hand, the BERT models excel in comprehending context-heavy texts, but struggles with higher-level abstraction and inference tasks due to a lack of explicit semantic knowledge, thus necessitating inference for unspecified relationships. The paper recommends further research on deep learning models for ontology alignment and merging. Also, the ensembling of deep learning models and the use of domain-specific knowledge for ontology learning require further research for ontology construction.

INDEX TERMS Deep learning, ontology construction, ontology learning, term extraction, relation discovery, axiom learning.

I. INTRODUCTION

Ontology Learning is a process of extracting and transforming data from various sources into a machine-understandable format automatically [1]. The ontology learning process

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

involves building a vocabulary from various sources by extracting terms, concepts, relations, and axioms and performing ontology alignment and merging [2]. Therefore, the terms ''ontology learning'' and ''ontology construction'' are used interchangeably in the rest of this review study. Ontology learning can be performed from different data sources, among them are plain text, semi-structured schemas, knowledge bases, dictionaries, relational schemas, web content, and social media data [3], [4]. Term extraction, relation discovery, and axiom learning are the three primary tasks in the ontology learning process [5], [6], where term extraction, involves identification and extraction of terms and concepts, along with their corresponding similarities; relation discovery, encompasses the extraction of attributes or relations and their classification; and axiom learning, entailing the extraction of axioms, rules, and the generation of predictions [3], [4], [7], [8]. Ontologies facilitate information sharing and knowledge representation, which is very important in this digital era, thus making ontology learning a crucial research area.

From the onset ontology construction was being done manually and the process was labor-intensive, costly, and difficult to cope with the ever-growing data on the web [9]. Manual ontology construction could not sustain the unlimited growth of data in terms of processing and knowledge representation. The increasing complexity of knowledge representation tasks as a result of vast data on the web [2] often referred to as ''big data,'' holds immense potential for uncovering valuable insights. However, traditional data analysis methods struggle to handle the sheer volume and complexity of this information [10], research was initiated on how machine learning techniques can be used for semi-automation of ontology construction [11]. The most common machine learning techniques are: Linear Regression, Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), Naive Bayes, k-Nearest Neighbors (kNN), Neural Networks, Gradient Boosting Machines (GBM), Principal Component Analysis (PCA), K-means Clustering Hierarchical Clustering, Association Rule Learning, Ensemble Learning, Genetic Algorithms, Reinforcement Learning, Hidden Markov Models (HMM), Gaussian Mixture Models (GMM), Autoencoders, Self-Organizing Maps (SOM) among others. Supervised machine learning algorithms were adopted in ontology construction mainly for classifying entities in a domain using labeled datasets and existing knowledge, while unsupervised machine learning approaches aid in pattern and relationship discovery without labeled data. However, Machine learning techniques exhibited many limitations in the field of ontology construction: (1) machine learning models are limited when applied to complex real-life situations such as speech, natural language, and images, (2) supervised machine learning algorithms require a significant amount of manual labeling and data validation, which consumes a lot of labor and time because training data must be generated for each target word to be disambiguated [10], (3) for relation extraction, human annotation and labeling is required and is difficult to implement with a large corpus [11], and (4) Machine learning techniques were not able to cope with large datasets [10]. Due to these challenges, there was a shift towards Deep learning techniques for ontology construction.

The primary objective of deep learning is to dig deep into the heart of data, decoding its complex characteristics and transforming them into semantically abstracted, higher-level dimensional representations [12]. The journey undertaken by deep learning models is a significant step forward, augmenting conventional machine learning models [13]. The machine learning techniques, while undoubtedly valuable, do possess some limitations when it comes to the representation and encoding of voluminous textual content [2]. This limitation, in turn, restricts their ability to produce relations between the rich textual information and the complex web of ontology concepts [2], [14].

Deep learning, on the other hand, exhibits a remarkable capacity to encode complex notions such as word dependencies, context, and the sequencing of words through the utilization of vector representations [14]. This complex process leads to the creation of great enriched embeddings for the input data, revealing new areas of understanding and insight in the ever-changing view of data analysis and knowledge representation.

There has been notable progress in researching the use of deep learning techniques for ontology learning. The scope of the work in this field is vast and encompasses a diverse array of tasks, ranging from word embedding and term extraction to the more complex practices of relation discovery and axiom learning [13]. The future of deep learning in this area appears to be on a trajectory of continued and remarkable advancements. This projection is largely driven by the unique advantage that deep learning holds, namely its capacity to minimize the need for manual engineering [15]. This distinct character empowers deep learning to effectively leverage available computational resources and data [15], thereby paving the way for more efficient and scalable ontology learning solutions [14].

The literature presents an outstanding gap in deep learning for ontology construction, [2], [14] absence of a comprehensive and in-depth review that explains how deep learning techniques are used across various stages in ontology construction. This paper presents a significant contribution of deep learning for ontology construction. The main aim is to bridge this knowledge gap by availing a detailed review of the application of deep learning techniques in ontology construction, shedding more light on the current situation, and discussing promising directions and potential avenues for future improvements in the use of deep learning for ontology construction. The next section expands on the topic of Deep Learning further.

II. DEEP LEARNING

Deep learning finds its roots in a specific type of artificial neural network (ANN) [18]. Specifically, it is a computational

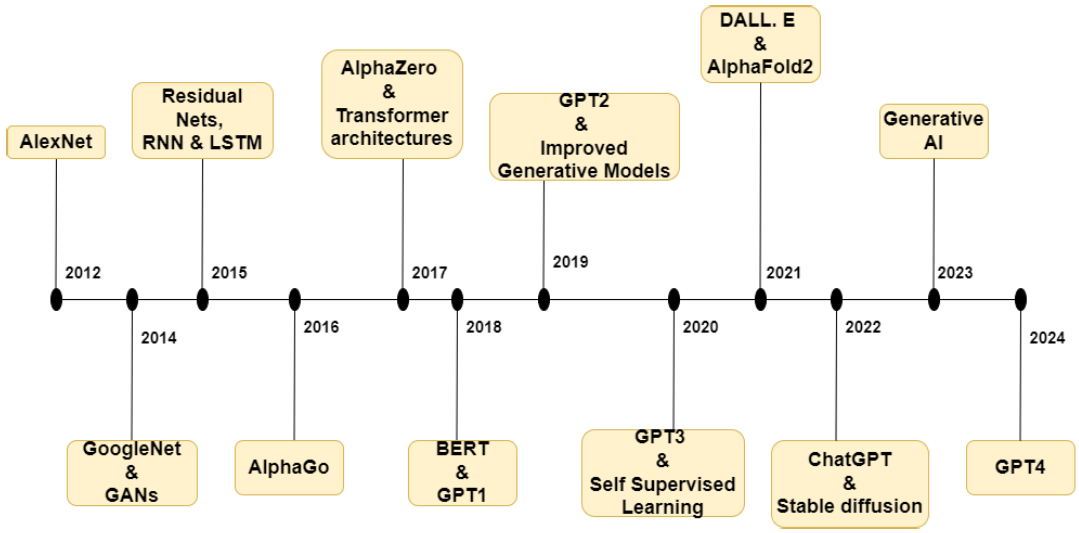


FIGURE 1. Evolution of Deep Learning architectures from 2012 to 2024. [16], [17].

model inspired by the structure and function of the human brain. ANNs comprises interconnected processing units called neurons, arranged in layers. These layers include an input layer for receiving data, hidden layers for processing information, and an output layer for producing desired outputs [18]. Deep learning derives its name from the depth of these hidden layers, allowing for the extraction of complex patterns and representations from the input data. The layers in the deep learning models are capable of discovering more complicated abstractions of the input information.

A. EVOLUTION OF DEEP LEARNING ARCHITECTURES

Many deep learning architectures exist in literature. Figure 1 shows the various deep learning architectures launched starting from 2012 where AlexNet, a groundbreaking convolutional neural network (CNN) was introduced by Krizhevsky et al. [19] achieving significant performance in the ImageNet Challenge, marking a shift in computer vision. GoogleNet/Inception was launched in 2014 and it introduced ''inception modules'' to enhance computational efficiency in CNNs, winning the 2014 ImageNet Challenge. The Generative Adversarial Networks (GANs) [20] was also launched in 2014. The GANs are deep neural net architectures composed of two nets, pitting one against the other (thus the ''adversarial''). They can learn to mimic any distribution of data and can generate content in domains such as images, music, speech, and many more.

In 2015, ResNet was introduced [21] to address training difficulties in deep networks with residual connections, achieving state-of-the-art results across various computer vision tasks. Similarly, an improved version of LSTM networks, which effectively process sequential data and mitigate the vanishing gradient problem in recurrent neural networks, was launched in 2015. Although the original concept, dates back to 1997 with the advent of neural computation [22]. AlphaGo was introduced in 2016 [23] the world's number one Go player, and later upgraded in 2017 to AlphaZero [24] which learned how to play master-level chess in just four hours and defeated Stockfish (the top AI chess player) in a 100-game match — without losing a single game.

BERT was launched in 2018 [25]; it is a deep learning architecture which revolutionized natural language processing by pre-training bidirectional text representations, achieving state-of-the-art results across NLP tasks. From 2018-2024 a series of GPT models were introduced [26], [27], [28]. The GPT models are large-scale transformer models for natural language understanding and generation, they demonstrated remarkable capabilities in text-related tasks. Also DALL-E was launched in 2021 [29]. It is a variant of GPT architecture specifically designed for image generation from textual descriptions.

The papers reviewed in this study used various Deep Learning architectures and their ensembles including CNNs, RNNs, LSTM, GRU, BERT and GPT, to address the scope of the study that relates to operations such as concept extraction, relation extraction, axiom learning, data classification, alignment and merging of ontologies on textual data, all which are crucial tasks of ontology construction.

B. DEEP LEARNING IMPLEMENTATION

Implementation of deep learning models involves critical stages which include data collection, model training, and performance evaluation [30]. Figure 2 shows the implementation of deep learning for the detection and classification of plant diseases. Initially, the dataset is gathered and subsequently divided into two segments, the training and the validation. The deep learning models are then trained either from scratch or by using transfer learning technique, then the training or validation progress is visualized to gauge their effectiveness [30]. Performance metrics are then

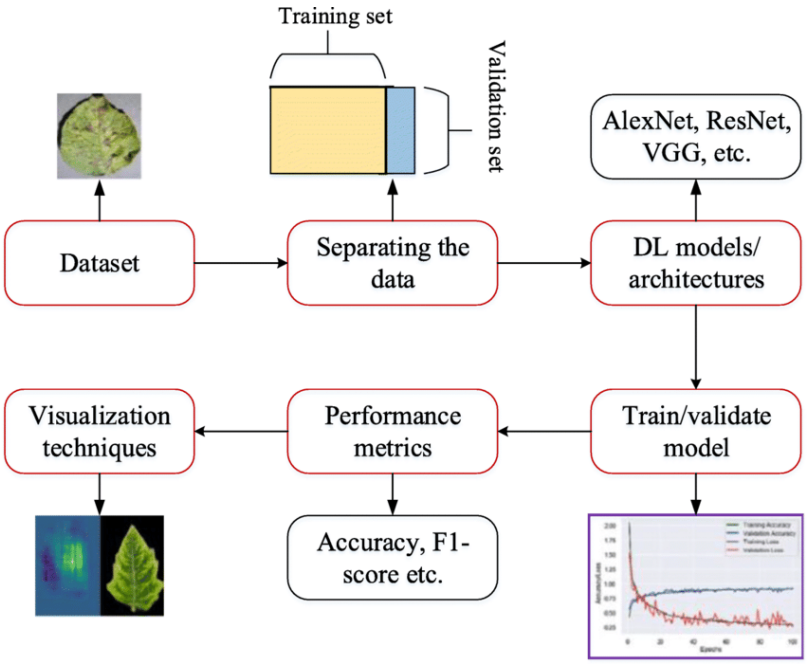


FIGURE 2. A complete flow of Deep Learning Implementation [30], [31].

employed for image classification that is identifying specific plant diseases leading to the utilization of visualization techniques or mappings for image detection, localization and classification [30], [31]

III. METHODOLOGY

This section outlined the procedure used for the selection, collection, and review of papers on this study. To ensure transparency in our selection process, we detail the search keywords, techniques, data sources, databases, and the inclusion and exclusion criteria employed to identify relevant research papers, specifically focusing on the applications of deep learning in ontology construction. The systematic literature review procedure detailed in the work by Weidt and Silva [32] was followed, and we were guided by Jauro et al. [33].

A. SEARCH KEYWORDS

With a clear understanding of our review goals, we selected keywords to ensure we retrieved the most relevant articles. Initially, several keywords were formulated and later narrowed down based on the research objectives. The list of keywords used for searching the articles include ontology learning, ontology construction, deep learning, term extraction, relation extraction, axiom learning, machine learning, classification using deep learning, deep learning and ontology construction, deep learning and ontology learning, convolution neural network, Bidirectional Encoder Representations from Transformers for relation extraction, generative pre-trained transformer, recurrent neural networks, deep reinforcement learning, deep neural network in ontology construction, and many more, the keywords were utilized to search relevant academic databases to obtain the articles.

B. SEARCHING THE ARTICLES

To ensure our search captured the latest advancements, we conducted it in two phases. The initial search occurred between August 26th and 30th, 2023, while the final search took place between December 13th and 19th, 2023. Articles retrieved based on keywords were further scanned on each search to obtain more related articles from their citations and references sections.

C. ACADEMIC DATABASES

Building on the keywords identified earlier, we focused on retrieving articles from credible sources. These included peer-reviewed journals, edited books, and conference proceedings indexed in various academic databases The following are the academic databases that we used to extract the articles: Springer, IEEE Access, Google Scholar, Elsevier (ScienceDirect and Scopus), ISI Web of Science, DBLP and Wiley Online Library. These repositories house high quality publications from prestigious SCI-indexed journals and top international conferences.

D. ARTICLE INCLUSION/EXCLUSION CRITERIA

We evaluated each paper based on its title, abstract, conclusions, and even the full content, to determine if it directly aligned with our research goals. To ensure we focused on the most relevant articles, we established clear inclusion and exclusion criteria detailed in Table 1.

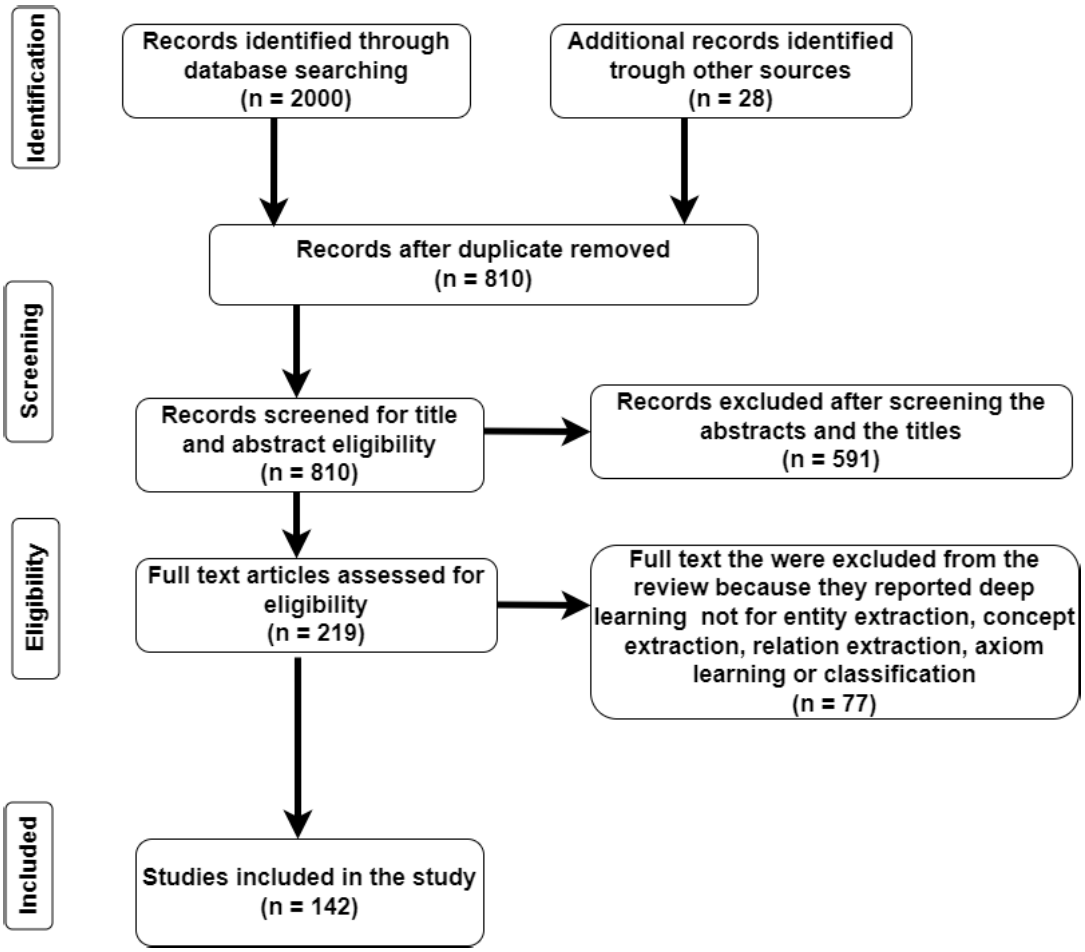


FIGURE 3. Prisma diagram showing the search and selection process.

TABLE 1. Inclusion and Exclusion Criteria.

Inclusion	Exclusion
The review focuses on only deep learning architectures	Articles on traditional/ shallow machine learning were not considered
Articles with the application of deep learning in ontology construction tasks were considered	Articles with the application of deep learning in non-ontology construction tasks were not considered
Articles published in reputable peer review journals, conference proceedings, and edited books were considered	articles published as part of text books, abstracts, editorials and keynotes speeches were excluded
Only articles written in English language were considered for the review	Articles written in other languages were not considered for review

After application of the inclusion/exclusion criteria in Table 1, a total of 2028 papers were obtained from the whole academic databases at the initial search. A number of 1 818 articles were eliminated after removing duplicates and implementing elimination based on titles, abstracts, and conclusions. On the second phase, full text was assessed where a total of 77 articles were rejected. 142 papers retained and hence were used for the review. Figure 3 shows the article selection process following the guidelines for reporting from PRISMA [34]. The use of Deep Learning in various operations of ontology constructions is discussed in the next sections.

IV. PREPROCESSING AND WORD EMBEDDINGS

The text preprocessing stage involves collecting text from various sources and transforming it into a format that can be readily understood [35]. Text transformation enhances the workability of the input text, facilitating its subsequent analysis and prediction during the deep-learning phase. Text preprocessing involves two components which are text normalization and natural language processing (NLP). NLP tasks include tokenization, ensuring token uniformity, removing diacritics, purging extraneous characters, and segmenting the text into distinct sentences [10].

Albukhitan et al. [12] introduced a Word2vec model for word embedding. The model integrates the Continuous Bag of Words (CBow) and Skip-gram algorithms [36]. The model uses brief word representations, simplifying the training process, when dealing with extensive datasets [37], in contrast to complex deep neural network architectures [38]. The model has shown exceptional performance in tasks

related to word representation, authors collectively label this combination of CBow and Skip-gram models as deep learning models [39].

The CBOW a feed-forward neural network with three layers was proposed by Mikolov et al. [37], as shown in Figure 4 with an example sentence were the word ''love'' is of interest.

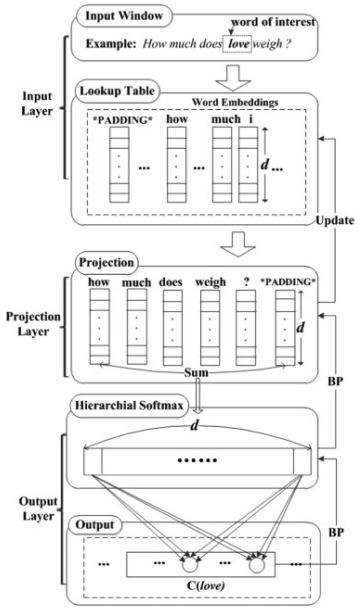


FIGURE 4. Architecture for CBow model. [40].

The model accepts a sentence as its input, for example, the phrase ''How much does love weigh?'' the input layer provides embeddings for individual words, initialized as random, dense float vectors with d dimensions. The specific word under examination is employed to create word embeddings embedded within the specific sentence [37]. The embeddings are then integrated into the projection layer. The projection layer then consolidates all these embeddings. Ultimately, an output layer employs a frequency-based Huffman tree for word encoding. This architectural strategy endows CBow with a notably high level of efficiency [40].

In the skip-gram variation, the objective is to identify probable words likely to appear before or after a specific word in its linguistic context [41].The window size defines the extent of the textual neighborhood examined for contextual information around a target word, rather than indicating the count of context words [40]. Suppose we have the following sentence ''Tsitsi, a tall brown girl is walking down the road''

Considering the sentence ''Tsitsi, a tall brown girl is walking down the road'' in Table 2. The word ''tall'' is selected as the context word. Then, the relationships between the target word and its neighboring words is analysed by generating pairs within a predefined distance. In this example, the words 'Tsitsi', 'a', 'girl' and 'brown' are within the window. The first pair consists of ('tall'; 'Tsitsi'), the second pair consists of ('tall'; 'a'), the third pair consists of ('tall', 'brown'), and the fourth pair consists of ('tall'; 'girl') as shown in the right cell of the first row of Table 2 [42].

TABLE 2. How context-target word pairs are produced when training the Skip-Gram model [41].

Window size	Text	Skip-grams
2	[Tsitsi , a tall brown girl] is walking down the road	[tall, Tsitsi] [tall, a] [tall, brown] [tall, girl]
2	Tsitsi , a [tall brown girl is walking] down the road	[girl, tall] [girl, brown] [girl, is] [girl, walking]
2	Tsitsi , a tall brown girl is walking [down the road]	[road, the] [road, down]
3	[Tsitsi , a tall brown girl is walking] down the road	[brown, Tsitsi] [brown, a] [brown, tall] [brown, girl] [brown, is] [brown, walking]
3	Tsitsi , a tall brown girl is [walking down the road]	[road, walking] [road, down] [road, the]

Adewumi et al. [38] and Mikolov et al. [37] agreed that CBow and Skip-gram performed better when training word representations using a large dataset. The resultant vectors produced proved to be more effective in identifying word similarities, both syntactically and semantically, achieving cutting edge results [43]. Leimeister and Wilson conducted an experiment [36], that explored word embedding learning within a hyperbolic space through the hyperbolic skip-gram model and obtained good results. Another study by Albukhitan et al. [12], introduced a blueprint for word embedding by employing CBow and SKIP-G deep learning models. The overall performance of this approach showed great promise.

On the contrary, CBow relies on a series of preceding and subsequent words to correctly identify the central target word. The model analyzes the context around a specific word in a sentence by averaging the vector representations of subsequent words within a defined window. As a result, the arrangement of words before or after the target word does not affect the resulting mean vector. While the continuous skip-gram technique seeks to predict words within a defined range by classifying terms preceding and succeeding the central word. The skip-gram approach prioritizes nearby words over those at a distance [38]. Although the model simplified

the training of word embeddings, it presented weaknesses in recognizing non-taxonomic relation extraction. Also, it failed to annotate a range of documents from various domains, which could have been used to assess the ontology learning process across these domains [39].

Ayadi et al. [35] introduced a deep bidirectional language model (BiLM) capable of being pre-trained on a big dataset to generate word embeddings. A Bidirectional Language Model (BiLM) combines the forward and backward language models to improve the log-likelihood of θ in both directions to analyze a sequence of N tokens $(t_1, t_2, \ldots, t_N)$ as shown in the following expression $\sum_{k=1}^{N}(logp(t_k|t_{k+1}\ldots\ldots t_N;\overrightarrow{\theta})+logp(t_k|t_{k+1}\ldots\ldots t_N;\overrightarrow{\theta}))$ where $\overrightarrow{\theta}$ and $\overrightarrow{\theta}$ are the parameters of the forward and backward language models respectively. The research demonstrates that each BiLM layer contributes to representing intricate syntactic and semantic relationships between words within a sentence. Utilizing all layers fosters a richer understanding of language, resulting in enhanced model performance. Peters et al. [44] introduced ELMo, a pre-trained language representation, which adopts a feature-based approach. Contextual representations of words are formed by aggregating information from all layers. A model for a specific task involves integrating these layers through a weighted combination [45], using max-pooling on the language model states and adding a softmax layer text for classification. Radford et al. [46] worked on language representation through the Generative Pre-trained Transformer (GPT), employing a fine-tuning approach. GPT employs a semi-supervised learning methodology to encode language patterns by utilizing transformer decoders [46]. Primarily designed for text representation, GPT comprises 12 transformer layers and 12 attention heads within its transformer decoder architecture. It capitalizes on extensive unlabeled datasets, notably the BooksCorpus dataset [47], by pre-training on them and subsequently fine-tuning on more limited supervised datasets.The primary function of GPT is to forecast the subsequent token within a sequence, a task accomplished through the architecture depicted in Figure 5.

Figure 5 represent a GPT-2 model. This architecture comprises two key parts: the encoder and the decoder [49], [50]. The encoder uses self-attention and a feed-forward layer to shrink input text, while the decoder adds encoder-decoder attention for aligning results [46]. This concurrent processing arranges memory and handles long dependencies. GPT-2 takes word vectors, predicts the next word's likelihood, and appends it to the input sequence for the next step [50], [51]. Both of these approaches are unidirectional, which limits the range of architectures available for pre-training.

GPT's input consists of the weighted embeddings of input texts along with their positional embeddings, facilitating context extraction. This input undergoes processing through the multi-head attention layer within the 12-layered transformer decoder blocks, followed by a feed-forward layer, culminating in softmax outputs representing a probability

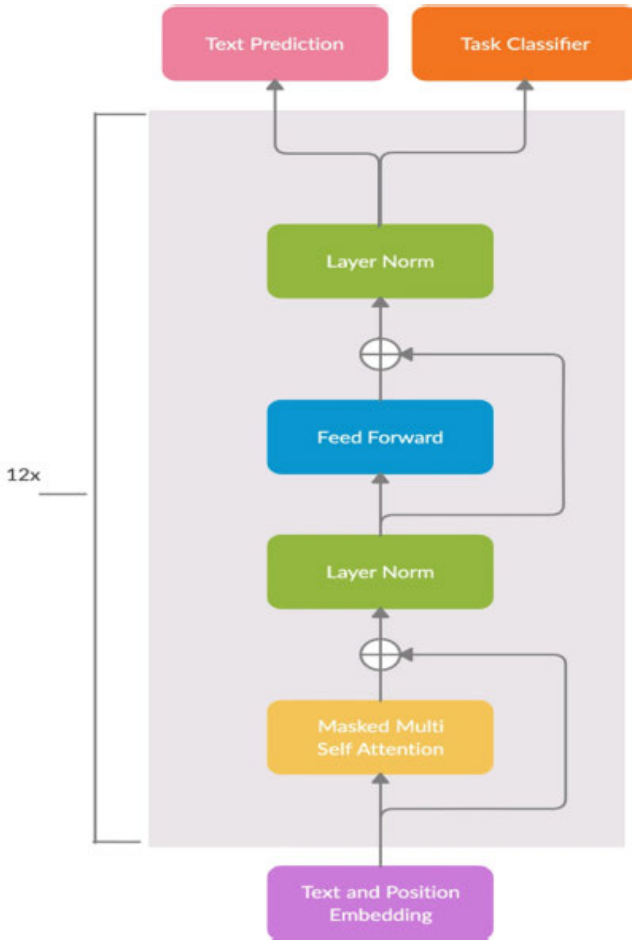


FIGURE 5. Architecture for GPT-2 model. [48].

distribution. Mathematically, this process can be expressed as follows:

$$h_o = UW_e + W_p \tag{1}$$

$$h_l = transformer_block(hl_1)\forall x \in [1, n] \tag{2}$$

$$p_{(u)} = softmax(h_n * W_e^T) \tag{3}$$

where u = (u-k,.....u-1) is the context vector of tokens, $k$is the context window size, n is the number of layers, W_e is the token embedding matrix, and W_p is the position embedding matrix. The GPT model can be pretrained and fine-tuned for other tasks. Scaling up the model has produced the GPT-2, GPT-3 and GPT-4 architectures. In 2020, GPT-3 was launched [28], boasting a much larger size (175 billion parameters) trained on a vast dataset and could handle different tasks well, even in zero-shot and few-shot settings across various domains. Brown et al. [28] illustrated that enlarging language models can significantly enhance their ability to perform tasks without specific training, reaching a level of competitiveness comparable to previous state-of-the-art fine-tuning methods. However, there were concerns about generating inappropriate or biased content by these ChatGPT models [52], [53].

According to the OpenAI report, [26] the latest version, GPT-4, a multimodal model which can take image and text as inputs and produce text outputs, was launched and it continues the trend of making sure the model aligns with human values by incorporating RLHF [26]. GPT-4 can solve problems and understand language without specific instructions, making it a potential candidate for future Artificial Intelligence systems. Advancements like ChatGPT Plus, built on GPT-4, further demonstrate the potential for these models to be incredibly helpful tools. Overall the evolution from GPT to GPT-4 showcases the transformative power of large language models while simultaneously tackling issues related to bias and alignment [54].

To improve fine-tuning-based methods, Devlin et al. [25] introduced a groundbreaking language representation model known as Bidirectional Encoder Representations from Transformers (BERT), designed for text preprocessing and word embedding. BERT is offered in two configurations: Base, with a smaller parameter count suitable for lighter tasks, and Large, with a larger parameter count for demanding tasks. The $BERT_base$ variant comprises 12 layers, 768 hidden dimensions, and 12 attention heads, resulting in a total parameter count of 110 million [55]. While $BERT_Large$ has 24 layers, 1024 hidden dimensions, 16 attention heads, and a total of 340 million parameters [55]. According to Devlin et al., [25], the BERT model combines components from both feature-based and fine-tuning approaches in its language representation strategy. BERT utilizes masked language models to pre-train deep bidirectional representations, meaning it learns word meaning by predicting masked words while considering both preceding and following words in a sentence. As depicted in Figure 6, BERT's methodology unfolds in two stages: pre-training and fine-tuning.

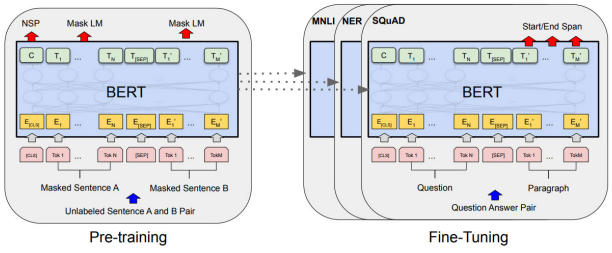


FIGURE 6. Architecture for BERT model. [25].

Devlin et al., [25] describe BERT's training as a two-stage process. First, the model pre-trains on a large corpus of unlabeled text using tasks like predicting masked words and understanding sentence relationships. Then, for specific tasks like sentiment analysis, the pre-trained model is fine-tuned on labeled data [25]. Resolving the limitation of unidirectionality, the BERT model introduces the concept of a masked language model (MLM). The approach combines word and sentence representations within a comprehensive pre-trained transformer [44], using masked language modeling to predict the next sentence. BERT uses advanced techniques to obtain competitive results and proves well-suited for feature-based and fine-tuning methodologies [25].

Shang et al. [56] proposed a G-BERT model for Medication Recommendation, the model combines Graph Neural Networks (GNN) and the BERT model for word embeddings. G-BERT produces the initial embedding for medical codes by utilizing graph neural networks with a medical ontology, G-BERT develops a flexible BERT model for representing single-visit. The model is fine-tuned for the medication recommendation task [57] in the prediction layer. In an experiment by Jeong et al. [58], a novel method for enhancing citation recommendation tasks using paper citation graphs was introduced. The model integrates the results obtained from Graph Convolutional Networks (GCN) and BERT to improve predictive tasks downstream. The study emphasizes that integrating fine-grained local information with a broader global understanding is critical for achieving positive outcomes. This multi-layered approach transmits valuable information from the GCN to the input text and distributes the text's representation throughout the GCN, leading to a more nuanced understanding.

V. TERM EXTRACTION

Term extraction involves identifying and extracting terms and concepts, along with their corresponding similarities. A study by Wang et al. [59] employed a Deep Belief Network (DBN), a type of advanced artificial neural network, to extract information about academic activities from unstructured documents.

Also, another study by Zhong et al. [60], [61] employed a Deep Belief Network (DBN) for the purpose of unsupervised concept extraction. Firstly the named entities were identified from the conditional random field (CRF), and then the extraction of the named entities was done using DBN. The DBN is a deep neural model comprised of multiple Restricted Boltzmann Machine (RBM) layers and a Back Propagation (BP) layer [59], [60], [62], [63] as shown in Figure 7.

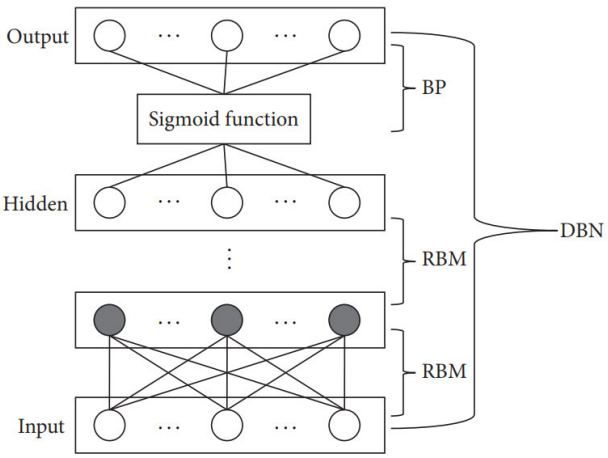


FIGURE 7. Structure of DBN. [59], [60].

The model can classify high-dimensional sparse feature vectors using its two-step training procedure. Initially, unsupervised training is performed on every layer of the restricted Boltzmann machine (RBM) where the first hidden layer (1) and the raw input feature vector $V(1)$ are sent into the first RBM layer. The second RBM layer then uses the output of the first layer as the input $V(2)$. Figure 8 illustrates how the unsupervised training process continues for the whole RBM network [64], [65], [66].

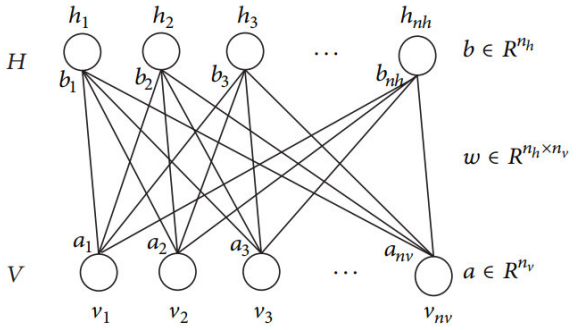


FIGURE 8. Structure of RBM. [59], [60].

The RBM architecture, depicted in Figure 8, comprises two distinct layers: the visible layer (V) containing the observed data and the hidden layer (H) responsible for extracting abstract representations, organized in an undirected graph [59], [60], [60], [67]. The RBM architecture comprises two distinct layers: the visible layer (V) consisting of n_v neurons (think of them as data input units) represented by the state vector V, and the hidden layer (H) with n_h neurons represented by H. Each layer has its bias vector (a for V and b for H) and connects to the other through a weight matrix (w). The RBM operates based on an energy function E in Equation (4)) that defines the state of the model (v, h).

$$E_\theta(v,h) = -\sum_{i=1}^{n_v} a_i v_i - \sum_{j=1}^{n_h} b_j h_j - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h} h_j w_{i,j} v_i \quad (4)$$

The vector can further be represented as in Equation (5)

$$E_\theta(v,h) = -a^T v - b^T h - h^T w_v \quad (5)$$

The RBM's energy function, defined in (5, measures how "good" a particular combination of visible and hidden layer states is. By applying a mathematical trick called the Boltzmann distribution (Equation (6)), we can determine the probability of each possible configuration.

$$P_\theta(v,h) = \frac{1}{Z_\theta} e^{-\theta}(v,h), Z_\theta = \sum_{v,h} e^{-E_\theta(v,h)} \quad (6)$$

where Z_θ denote the normalised factor. The activation of a hidden layer node h is conditioned on the activations of the visible layer node v, as expressed in Equation (7). This conditional relationship forms the basis for information propagation and feature learning within the RBM.

$$P(h_j|V) = \frac{1}{1 + exp(-a, -\sum_i v_i w_i)} \quad (7)$$

The symmetrical architecture of RBM allows for a bidirectional information flow were the hidden layer H extract features from the visible layer V and reverse the process by reconstruct the original data using Equation (8).

$$P(v_i|H) = \frac{1}{1 + exp(-b_i, -\sum_j h_j w_{j,i})} \quad (8)$$

During RBM training, the model transforms surface-level features in the visible layer into richer representations in the hidden layer. The process involves optimizing internal parameters to maximize the likelihood of both visible and hidden layer states occurring together. Additionally, the model aims to accurately reconstruct the original data by minimizing the discrepancy between the actual and reconstructed visible layers.

In second step (Figure 7), supervised training is conducted in the Back Propagation (BP) layer. The BP layer classifies features produced by the final RBM layer. In this phase, areas for improvement are highlighted and targeted feedback is sent to each RBM layer, ensuring resources are not wasted on unnecessary adjustments. The strategy leads to optimal performance with minimal effort [68], [69]. A list of recognized entities or concepts is the final result of the DBN-based concept extraction process. Tags and classification are applied to these entities depending on the application, and the results used for information retrieval, knowledge base collection, and question-answering. The DBN-based concept extraction process builds ontologies for storing and reasoning with knowledge. The DBN also classifies documents in a database, allowing for more accurate searching. The findings demonstrate that DBN is a successful method for extracting information in high-dimensional feature spaces. It surpasses groundbreaking learning models like Support Vector Machines (SVM) and backpropagation networks in terms of performance [70], [71]. The model's inability to consider the sequential nature of the training samples limits its capacity to capture temporal dependencies and extract exact features, ultimately impacting the effectiveness of classification.

In an experiment conducted by Arguello et al. [14], the objective was to automatically identify term variants for gene and protein names from a large corpus of biomedical publications, employing CBow and Skip-Gram techniques. The study comprised two distinct experiments. The first experiment aimed to ascertain whether the word embeddings generated by CBow and Skip-Gram could yield term variants for gene and protein names. The second experiment aimed at determining whether integrating domain ontology CVDO could enhance the discovery of term variants without modifying the underlying word embeddings. In first experiment, Skip-Gram achieved a success rate of 89% in obtaining term variants, while CBow achieved 64%. In the second

experiment, where the CVDO ontology was employed to provide biological knowledge, Skip-Gram demonstrated an even higher success rate of 95% in discovering term variants, compared to CBow's rate of 78%. This study showed that integrating domain ontology improved the deep learning models' performance.

Manda et al. [72] introduced a model that combines the Gated Recurrent Units (GRU) and Long Short Term Memory (LSTM), encoded GRU-LSTM, for automation of Named Entity Recognition (NER) of ontology concepts from text [73], [74]. The architecture of the GRU-LSTM model presented in Figure 9 uses CNN for character extraction in the text. The model accepts three inputs which are character embeddings (X_{tr}^{c}), word embeddings(X_{tr}^{w}), and part of speech embeddings (X_{tr}^{PoS}). The output from the CNN is combined with part of speech embeddings and ELMo word embeddings and becomes input to a 200-unit Bi-GRU. Leveraging two gating mechanisms, the Update and Reset gates, the GRU selectively retains and transmits information within its architecture [75]. The Update gate controls the incorporation of past knowledge into future states, while the Reset gate dictates how much historical information to discard. These vector-based gates regulate information flow, enabling the GRU to excel at learning and preserving long-term dependencies in sequential text [72]. The hybrid LSTM-GRU network can effectively mitigate

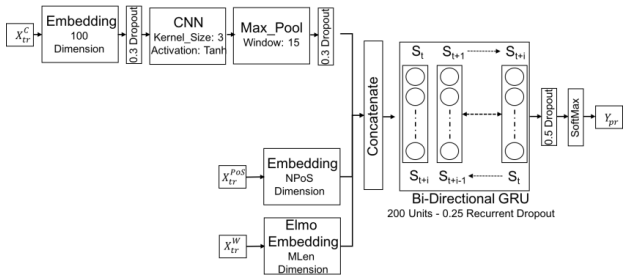


FIGURE 9. GRU-LSTM model [72], [73].

the vanishing gradient problem [72]. The LSTM's memory cell preserves long-term connections, while the GRU's gates regulate information flow, ensuring that the RNN remembers the most relevant data points throughout its journey [73], [76]. The forget gate's selective decisions ensure the LSTM unit retains only the most essential information, fostering long-term memory and learning as shown in Figure 10 [76]. The memory cell receives two inputs: the previous moment's output represented as h_{t-1} that provides context from past experiences, and the current moment's external information, x_t, represents new data to be processed [77]. A single long vector, $[h_{t-1}, x_t]$ produced using a transformation in Equation 9 by combining the inputs, ensures all relevant information is considered for decision-making [73], [76].

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{9}$$

where W_f and b_f stand for the weight matrix and bias of the forget gate, respectively, while σ is the sigmoid function. The

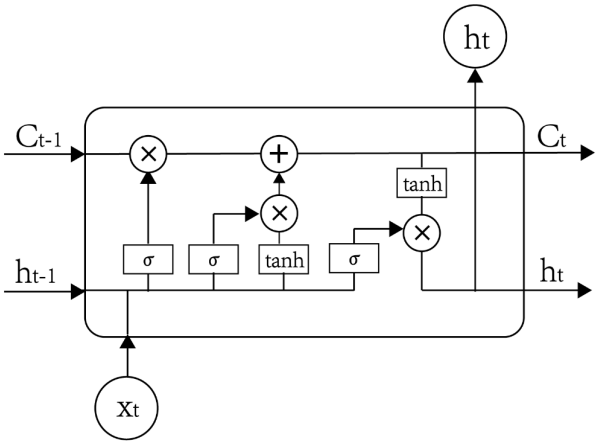


FIGURE 10. LSTM Architecture [76], [77].

forget gate examines the previous moment's output C_{t-1} and the current input C_t to determine relevant information [77]. It also assigns a value between 0 and 1 to each memory in the cell state where 1 signals "keep": essential information to retain and 0 signals "discard": outdated or irrelevant memories. The input gate identifies updates using the sigmoid layer (Equation 10) to highlight parts of the cell state that need refreshing. It also creates candidates by generating a new candidate vector of potential memories using a tanh layer (Equation 11). Therefore, the input gate filters out irrelevant content by ensuring only valuable additions enter the memory cells [76].

$$i_t = \sigma(W_t[h_{t-1}, x_t]) + b_f) \tag{10}$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{11}$$

$$C_t = (f_t * C_{t-1} + i_t * \tilde{C}_t) \tag{12}$$

where, x_t represent inputs, W_c the weights, h_{t-1} and C_{t-1} hidden state at the previous and current timestamp, b_c the bias, f_t the forget gate value, i_t the input, and C_t a new information value [76]. The output begins with an initial determination of output information using a sigmoid layer as in Equation 13. The cell state is passed through a tanh activation function, were the output of the tanh is multiplied by the output of the sigmoid layer to get the final output segment, as shown in Equation 14.

$$O_t = \sigma(W_\sigma * [h_{t-1}, x_t] + b_o) \tag{13}$$

$$h_t = O_t * tanh(C_t) \tag{14}$$

where, C_t represent a new information at timestamp t and O_t the output at timestamp t, The findings indicated that when the deep learning models are assisted with a domain-specific embedding model they produce better results [72], [73], [78]. Studies suggest that GRU is more efficient for extracting terms as it outperformed LSTM in achieving desired results.

Fan et al. [79] developed a sophisticated model, combining a multi-branch BiGRU layer and a Conditional Random

Field (CRF) model, for identifying geological hazard-named entities. The approach, detailed in Figure 11, allowed for effective entity extraction and even helped build a knowledge graph based on the extracted information [77]. The BiGRU model employs GRU units, which offer the advantages of LSTMs but with a simple structure shown in Figure 11 [79]. The GRU model in Figure 11 streamlines its structure with

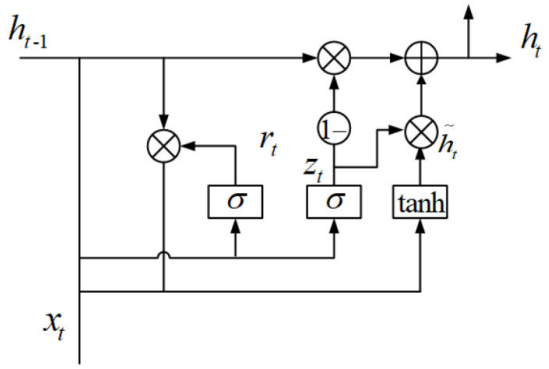


FIGURE 11. Structure of a GRU [77].

just two gates: Update gate (z_t) and Reset gate (r_t). The gates play crucial roles in regulating information flow within the model, as expressed mathematically in Equations 15 and 16.

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (15)$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (16)$$

where σ is the sigmoid activation function, h_t denotes the implied state and is defined in Equation 17.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (17)$$

where $\odot$ is the element product operator of two vectors and $\tilde{h}_t$ represents the candidate implied state and is defined in Equation 18.

$$\tilde{h}_t = tanh(W_{xh}x_t + W_{hh}(r_t \odot \tilde{h}_{t-1}) + b_h) \quad (18)$$

The candidate implied state, $\tilde{h}_t$, is dependent on the previous implied state, $\tilde{h}_{t-1}$, from previous time steps, which is controlled by the reset gate, r_t. The prior implicit state, $\tilde{h}_{t-1}$, is discarded when r_t gets near to zero. As a result, r_t functions as a method to decide how much historical information should be kept and to eliminate unnecessary prior implied states. The candidate implied state, z_t, and the prior implicit state, h_{t-1}, are updated by the implicit state, h_t, via the update gate. The model controls the weight of the previous implicit state in the present by varying the update gate. If the update gate is converging towards 1, it means the past state is preserved and transferred. The CRF model combines the maximum entropy and hidden Markov models. If each random variable Y_v adheres to the Markov property given a condition on the observation X, then the model is valid, this is defined in Equation 19.

$$P(Y_v|XY_u, u \neq v) = P(Y_v|XY_u, u \sim v) \quad (19)$$

where, (X, Y) forms a Conditional Random Field (CRF). where, X stands for the observed sequence, and uv includes all neighboring nodes of u connected to node v in graph G as in Figure 12.

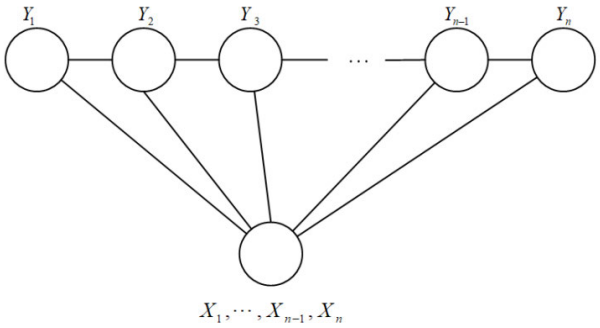


FIGURE 12. Structure of a CRF [77].

Lyu et al. [80] proposed a domain specific hybrid model of a pre-trained word embeddings and character-level representation model that combines Bidirectional LSTM and RNN (BiLSTM-RNN). Figure 13 shows the architecture of BiLSTM-RNN.

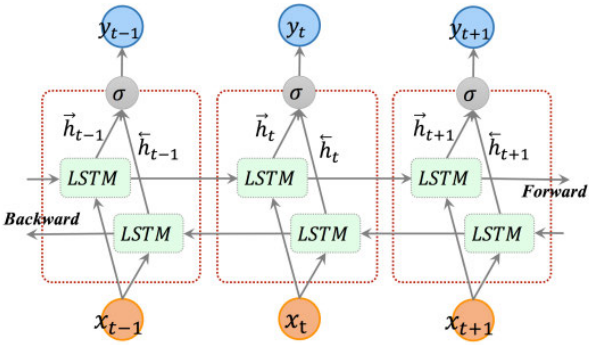


FIGURE 13. BiLSTM-RNN Architecture [80], [81].

The BiLSTM-RNN architecture processes sequential data in both forward and backward directions using separate hidden layers, connected to a shared output layer [82], [83]. The forward layer output sequence ($\overrightarrow{h}$) is computed chronologically from time $t - n$ to $t - 1$, while the backward layer sequence ($\overleftarrow{h}$) is processed in reverse order from $t - 1$ to $t - n$, where t represents the current time step, $t - 1$ the previous time step and $t - n$ a time step before the current time step t. This ensures the neural network considers past and future inputs when generating the output. The BiLSTM-RNN generates an output vector, denoted as y_t, where each element is computed using Equation 20.

$$y_t = \sigma(\overrightarrow{h_t}, \overleftarrow{h_t}) \quad (20)$$

where σ function is used to combine the two output sequences [84], [85]. A vector, $Y_t = [y_{t-n} \ldots\ldots y_{t-1}]$ represents the final output of a BiLSTM-RNN model. The BiLSTM-RNN architecture in Figure 13 was successfully

applied to perform Biomedical named entity recognition. Albukhitan et al. [12] argued that the results of the word embeddings produced by the language models can be used to extract relevant concepts by presenting a starting concept as an input to the ontology learning system. Thereafter, the similarity function can be used to find related concepts for each starting concept and the language model of all starting concepts.

Oksanen et al. [86] employed the BERT model for concept extraction. The BERT classifier considered the review sentence and the entity to be classified as an input [87]. By categorizing input sentences and entities, the classifier determines whether each is a feature aspect, product aspect, or non-aspect. Entity-associated tokens ('operating' and 'system') are replaced by a mask token during tokenization, as shown in Figure 14.

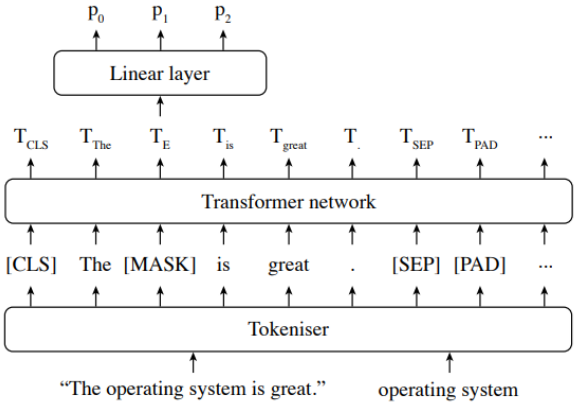


FIGURE 14. Architecture of the BERT classifier for aspect extraction [86].

The tokens are first processed by a transformer network, with a focus on the masked entity position T_E [88]. This allows the generation of probabilities, using linear layers and softmax operations, to determine if an entity is classified as a non-aspect, feature, or product. Additionally, the model provides the probability of an entity being an aspect of a product. A product aspect's likelihood can be calculated as in Equation 21.

$$p_0 = p_1 + p_2 \tag{21}$$

The process involves finding the average of the p_0 values for each entity from various input review sentences where it is the only entity. According to [87], when the average is greater than 0.65, the entity is classified as an aspect, but if the average of its p_2 values exceeds 0.45 its considered a product aspect. Instead of binary votes, the model utilizes raw output probabilities, which emphasizes the model's confident predictions during aggregation. The experiment showed promising overall performance [89].

The study by Fang et al. [90] utilized a deep learning-based model for the construction of a drought disaster ontology. The deep learning model integrated the BERT, BiLSTM, and CRF models to identify named entities from unstructured drought disaster documents. The integration of the deep learning models is shown in Figure 15. The BERT-BiLSTM-CRF model takes in the input texts on drought and calamity, where the BERT model then computes the word vectors. The BiLSTM layer identifies contextual features from the literature regarding drought disasters and generates the tag score. Lastly, the CRF layer puts constraints on the interrelationships of the tags for determining the perfect tag sequence, which is utilized in computing the matching tag for every entity influenced by the drought crisis. The model was able to extract low-level entities to create a knowledge graph for drought disaster management.

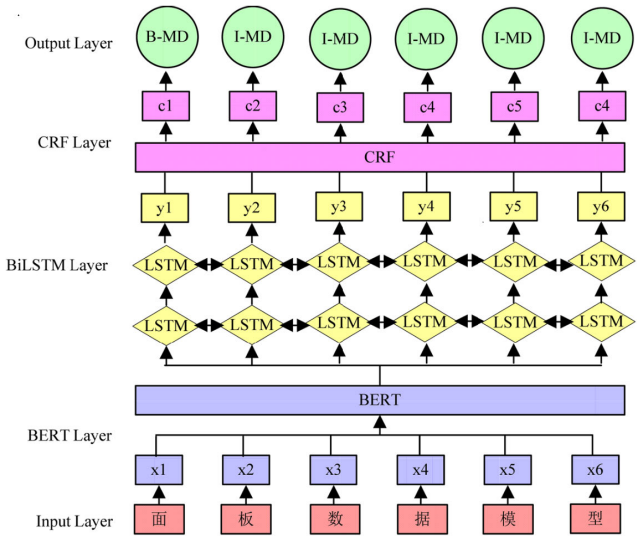


FIGURE 15. Architecture of BERT-BiLSTM-CRF model [90], [91], [92].

Fan et al. [79] employed the BERT-BiLSTM-CRF model to improve the reuse of geological hazard literature and serve as a valuable reference for geological hazard governance. Other studies by Jinjun et al. [91], [92] also explored the BERT-BiLSTM-CRF model for entity recognition to identify entities in autonomous transportation systems. Similarly, Liu et al. [93] harnessed the BERT-BiLSTM-CRF model for extracting entities on customer voice consultation questions, whereas, Ren et al. [94] implemented the BERT-BiLSTM-CRF model for extracting legal facts from Chinese legal texts; the model successfully identified complex entities from legal text details and mapped them into well-organized legal facts using the Chinese legal text ontology. Across all these studies that employed the BERT-BiLSTM-CRF model, the results demonstrated the models' efficiency and effectiveness in terms of extracting concepts.

VI. RELATION DISCOVERY

Relation discovery in ontology learning involves attribute or relation extraction and classification [1]. Relation extraction is done at different levels including sentence level, bag level, and document level, using supervised models, conventional neural models, or pre-trained models [95].

Ganesan et al. [96] employed an On2Vec model for relation extraction, which makes use of a two-component model, the Component Specific Models (CSMs) and the Hierarchical Model (HM). The CSMs outline concepts and relationships in a low-dimensional embedding space without losing relational features such as symmetry and transitivity. The authors [96] argued that a better way to understand hierarchical relationships is through an HM model. The results of the study show improved performances of relation extraction models. Furthermore, Ganesan et al. [96] proposed the use of similarity measures such as CBOW and cosine of Skip-gram, to quantify how closely two terms obtained from a language model are related to each other [37], [97].

Another study by Lamurias et al. [98] proposed a BO-LSTM model for relation extraction and classification. The BO-LSTM model uses domain-specific ontologies by representing each entity as a sequence of its ancestors in the ontology. BO-LSTM is implemented as a recurrent neural network with long short-term memory units and uses open biomedical ontologies, specifically Chemical Entities of Biological Interest (ChEBI), Human Phenotype, and Gene Ontologies to extract and classify relationships in text [99], [100], [101]. The results indicated that the BO-LSTM model improved the F1-score of both detection and classification of drug-drug interactions, in a document set with few annotations [102]. Another study by Sousa and Couto [103] used the BiOnto system to extract relationships in the biomedical field. BiOnto is a biomedical relation extraction system that uses bidirectional LSTM networks, advanced Word2Vec word embeddings, and multiple input channels to achieve optimal performance [97]. It also uses four domain-specific ontologies, with word embeddings and WordNet [97]. BiOnt can explore indirect ontology entries and populate its knowledge base with gold standard relationships, resulting in valuable evidence of previously unknown connections between biomedical entities [104]. The study suggests that using external knowledge sources, including biomedical ontologies, can improve entity relationship identification. Sousa and Couto [103] and Lamurias et al. [98] concurred that domain-specific ontologies improve the performance of deep learning techniques for biomedical relation extraction by facilitating the discovery of related concepts and also finding unknown relationships between biomedical entities [102], [105], [106].

Zeng et al. [107] conducted a study that utilizes a convolutional neural network (CNN) for relation extraction. Two distance embedding vectors were employed to represent the separation between each word and the two entities, while pre-trained word embeddings represented the tokens in a sentence. Sentence-level feature vector was extracted using CNN and max-pooling operations and made input into a feed-forward neural network with a softmax activation layer for classification [108], [109]. The model achieves an F1-score of 82.7%. Related studies by Wang et al. [110], Kim [111], and Bangyal et al. [112] all used CNN models to classify text. CNNs are multi-layer networks with convolutional and subsampling layers of multiple 2D layers and fully connected hidden layers [113]. In particular, CNN has an input layer that directly receives two-dimensional objects, a feature extraction process done through convolution, and the subsampling layer implemented using multiple fully connected hidden layers as shown in Figure 16. It is reported that [113], correct hyperparameters

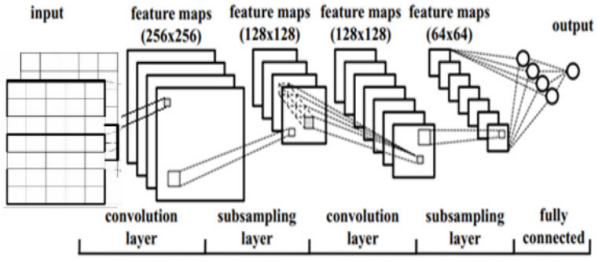


FIGURE 16. CNN architecture for text classification [112], [113].

setting improves performance during training. For instance, Guruvayur and Suchithra [113] developed a 6-step CNN algorithm for the classification of text and explained that, given a batch size of 50, running for 50 epochs, and setting the learning rate to 0.0001 produced better results. Another study by Bangyal et al. [112] studied Alzheimer's Disease and found that CNN categorized patient data into non-demented and demented groups. The dataset was divided into training, testing, and validation subsets, with adjustable model parameters for improved accuracy.

A study by Wang et al. [110] proposed a CNN model for classification of text and construction of a shipping business domain ontology. CNN was used for feature extraction and text classification to obtain word vectors for each category as shown in Figure 17.

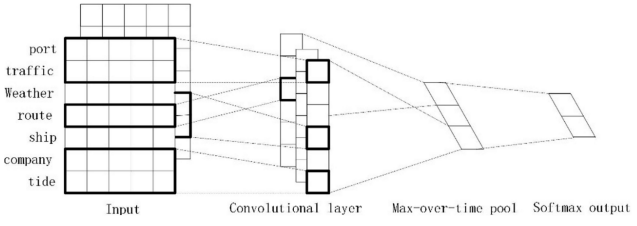


FIGURE 17. CNN architecture for feature extraction [110], [112].

A convolution operation uses a filter, called w, to create a new feature based on a window of h words in a word vector [114]. Using a window of words $x_{i:i+h-1}$, a feature c_i can be created, as in Equation 22.

$$C_i = f(w * x_{i:i+h-1} + b) \tag{22}$$

where, w are the filter weights, $x_{i:i+h-1}$ the window of words, and $b \in R^{n+h-1}$ a bias term. The function f is a non-linear function such as the hyperbolic tangent. This function is applied to every possible window of $x_{1:h}, x_{2:h+1}, \ldots x_{n-h+1:n}$

words in the sentence. The purpose of this filter is to generate a feature map, represented in Equation 23.

$$C = (C_1, C_2, \ldots\ldots C_{n-h+1}) \tag{23}$$

where $C \in R^{n+h-1}$. Max-overtime operation and dropout prevent overfitting and help the model generalize better. Dropout randomly drops out hidden units and is an effective way to prevent co-adaptation [115] and helps reduce the risk of overfitting and enables better performance on unseen data. The experiment demonstrated a high classification accuracy although the ontology was constructed with the assistance of experts. Similar studies undertaken by Kim [111], Fu et al. [116], Guo et al. [117], Li et al. [118] and Tian et al. [114], where, the authors employed convolutional neural network models for text classification, revealed that a simple CNN model with one convolution layer performs better. This finding was also supported by Bangyal et al. [112] who developed a CNN model for the construction of a domain ontology for Alzheimer's Disease.

Another study by Conneau et al. [119] demonstrated the use of a Very Deep Convolutional Neural Network (VDCNN) with 29 convolution layers for relation classification. The VDCNN model employs a lookup table to generate a 2D tensor (f_0, s) that contains character embeddings for s characters, meaning each character in the sequence is mapped to a high-dimensional vector representation that captures its meaning and context. Afterward, a layer of 64 convolutions of size 3 is applied to the tensor to extract local features and create a feature map that represents these features. Finally, the model employs blocks of operations to refine the feature map and extract higher-level representations of the input sequence. The VDCNN model builds a feature map for high-level text features, a convolutional encoder for deep hierarchical connections, and a k-max pooling for classification using three pooling procedures. ReLU units and softmax outputs are used to feed features into a three-layer classifier with 2048 hidden units. The findings showed that the classification results improved with the model depth [120]. The model suffers from high computational cost and memory requirements due to its deep architecture with a large number of parameters. However, this limitation can be addressed through techniques such as transfer learning where pre-trained VGGNet models can be fine-tuned on new tasks or datasets, making it more practical for various applications.

Bangyal et al. [112], [122] employed a deep convolution neural network (DCNN) for both feature extraction and relation classification. The DCNN model is comprised of three convolution layers of unique filters, i.e., 16, 32, and 64, with activation and max pooling layers coming after each convolution layer, followed by two fully connected layers with 512 neurons each and a softmax layer which does the classification using the sigmoid activation [121], the DCNN architecture is shown in Figure 18. Each word in the text is passed through the look-up table ($W_{v.d}$) to form the input document representation vector $n * d$, where n is the document's word count and d is its feature dimension. Once

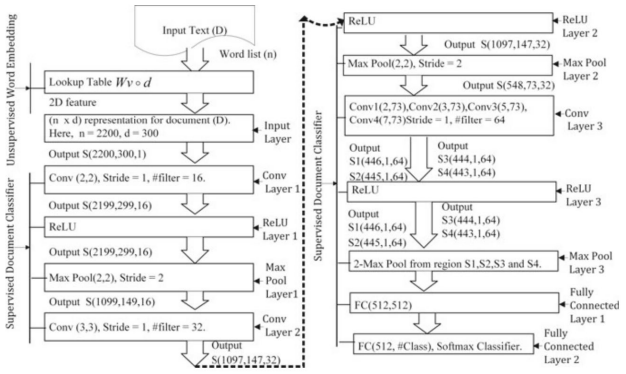


FIGURE 18. Architecture of DCNN. [121].

the input tensor extracts the local feature, the convolution layer learns the filter weights [121]. Furthermore, ReLU is used in deep convolution neural networks (DCNNs) to introduce nonlinearity and rectification layers for computation of gradient, mitigation of vanishing gradient problem, and reduction of computational time complexity [123]. Moreover, ReLU addresses the problem of gradient cancellation by assigning zero to negative values [124]. The pooling layer lowers the dimensions of the activation map, minimizes the risk of overfitting, and progressively lowers computing demands across the network. The softmax classifier is later applied to generate classification scores for each class as shown at the bottom right of Figure 18.

Nayak [125] proposed a multi-hop relation extraction using a hierarchical entity graph convolutional network (HEGCN) model. The method comprised two levels, the first level involves extracting entity relations within the document while the second level involves extracting entity relations across documents. The model improves the F1-score by 2.2% on a 2-hop dataset. The authors concluded that the model can be used for N-hop datasets. Furthermore, the study indicated that graph convolutional networks (GCNs) are capable of learning specific correlations among entities across documents [126], [127].

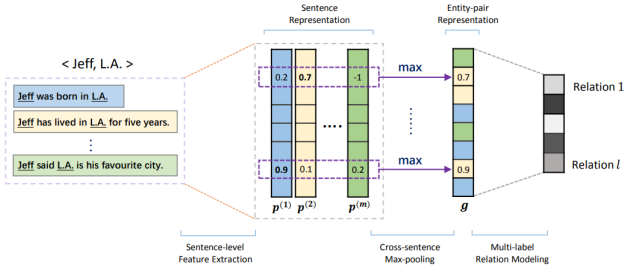


FIGURE 19. Architecture for MIMLCNN. [128].

In a study by Jiang et al. [128], the Multi-Instance Multi-Label Convolution Neural Network (MIMLCNN) approach for relation extraction was introduced. The method takes as input an entity pair (e_1,e_2) and the corresponding

sentence aligned with that pair, then generates a collection of knowledge bases that describes the relationship existing between these two entities. The model consists of three stages as shown in Figure 19. The first stage involves sentence-level feature extraction. It begins by padding the sentence length with zeros and converting it into a matrix representation. A convolution operation followed by a piecewise max pooling is applied to obtain a vector representation. The second stage, known as cross-sentence max-pooling, enables predictions based on evidence found within individual sentences and collectively across all sentences. The approach streamlines the direct extraction of relationships at the entity-pair level and allows for the compilation of evidence from various sentences. Max-pooling across sentences is chosen for entity-pair-level relation extraction due to its high efficiency since multiple instances of a feature do not provide significant additional information. The third stage is multi-label relation modeling a novel strategy for modeling distant supervision using neural network architecture. The label's confidence score for each sentence is calculated using Equation (24) based on the bias and a weight vector matrix.

$$o = W_{1g} + b_1 \tag{24}$$

where $W_1 \in \mathbb{R}^{3n*l}$ is a matrix of weight vectors for each label and $b_1 \in \mathbb{R}^l$ is the bias term. The probability of every connection is determined by a sigmoid function in Equation (25).

$$p(i|M, \theta) = \frac{1}{1 + e^{-o_i}}; i = 1, 2, 3, \ldots\ldots, l \tag{25}$$

where M denotes the set of the aligned sentences, and l is the number of relation labels. A binary label vector was then employed to signify the true relations included in the entity pairing. The method uses a shared entity-pair-level representation to handle the linkage between relationships. Two loss functions that are designed for multi-label modeling include sigmoid loss and squared loss defined in Equations (26) and (27), respectively.

$$loss\ sigmoid = -\sum_{i=1}^{l} y_i log(p_i) + (1 - y_i) log(1 - p_i) \tag{26}$$

$$loss\ squared = -\sum_{i=1}^{l} (y_i - p_i)^2 \tag{27}$$

where $y_i \in \{0, 1\}$ is the true value on label i.

Zhang and Wang [129] used a Recurrent Neural Network (RNN) for relation extraction as shown in the framework in Figure 20. The model in Figure 20 comprises a word embedding layer (bottom layer) where each word in a sentence is converted into a low-dimension word vector as follows. Let $X_t \in \{0, 1\}^{|V|}$ denote the one-hot representation of the t^{th} word v_t where $|V|$ denotes the size of the vocabulary V. The embedding layer transfers x_t to word vectors $e_t \in R^D$ as in Equation (28).

$$e_t = W_{e_m x_t} \tag{28}$$

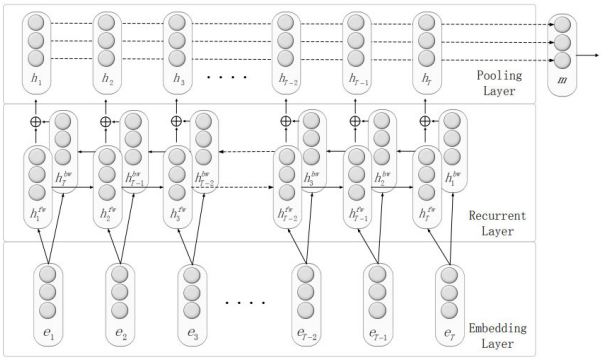


FIGURE 20. A framework for Recurrent Neural Network. [129].

where $W_{e_m} \in R^{|D|*|V|}$ is the projection matrix. Since x_t is one-hot, W_{e_m} stores the representation of all the words in V. The second layer of the model is the bidirectional recurrent layer (middle layer) which models the word sequence and produces word-level features. Consider a sequence $X = (x_1, x_2 \ldots\ldots x_T)$ representing the words projected into a series of word vectors $(e_1, e_2, \ldots\ldots e_T)$, with T the words count. The bidirectional recurrent layer receives the word vectors one at a time [129]. The output of the backward RNN and the forward RNN are then combined to create the prediction h_t at step t as in Equation (29).

$$h_t = h_t^{fw} + h_t^{bw} \tag{29}$$

where h_t^{fw} is the output of the forward RNN and h_t^{bw} is the output of the backward RNN. h_t^{fw} is defined by Equation (30).

$$h_t^{fw} = tanh(W_{fw} e_t + U_{fw} h_{t-1}^{fw} + b_{fw}) \tag{30}$$

where $h_t^{fw} \in R^M$ is the output of the RNN at the t^{th} step, M is the dimension of the feature vector, $W_{fw} \in R^{M*D}$, $U_{fw} \in R^{M*M}$, $b_{fw} \in R^{M*1}$ are model parameters. The hyperbolic function tanh() was chosen as the nonlinear transformation to facilitate error backpropagation because of its symmetry [129]. $h_t^{bw} \in R^M$ represents the output of the backward RNN and is defined in Equation (31).

$$h_t^{bw} = tanh(W_{bw} e_t + U_{bw} h_{t+1}^{bw} + b_{bw}) \tag{31}$$

where $W_{bw} \in R^{M*D}$, $U_{bw} \in R^{M*M}$, $b_{bw} \in R^{M*1}$ are the parameters of the backward RNN.The last (top) layer in the RNN structure in Figure 20 is the max pooling layer for producing a sentence-level feature vector by combining features from each word and selecting the highest value between the word-level features for each dimension. The sentence-level feature vector is then used in the relation classification process [129]. The architecture enhances the extraction of relationships between entity pairs while efficiently using information from sentences, making it a valuable tool for relation extraction tasks. Based on the experimental results on two different datasets, the model outperformed the CNN model proposed by Zeng et al. [122]

in relation classification and exhibited a powerful strength in detecting long-distance relation patterns.

Tang et al. [130], Meng et al. [131] and Li et al. [132] stressed the ability of LSTM to solve the challenges of vanishing gradient and the explosion problem. Furthermore, the authors in [133] argued that the Bidirectional LSTM (BLSTM), on the other hand, adds an extra hidden layer to the standard LSTM, which is connected to the initial hidden layer, therefore, making it able to consider information from both past and future contexts, facilitating the capturing of long-range dependencies in sentences. However, the BiLSTM fall short on extracting features. Convolutional Neural Networks excel in extracting local features though has got a weakness in learning sequential correlations [130], [131]. To overcome the limitations of both the BLSTM and CNN models, researchers have ensembled them, as shown in Figure 21. The integration allows the CNN model to identify local features, while the RNN model extracts long-term dependencies, providing a more comprehensive approach to the task.

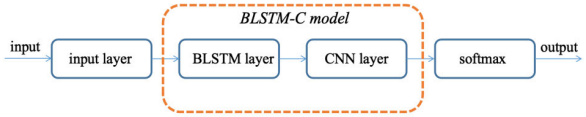


FIGURE 21. BLSTM - CNN [130], [132].

The input layer in Figure 21 plays a crucial role in pre-processing inputs, which involves removing words and segmentation. The results of the input pre-processing is fed into the BLSTM layer which generates the sequence output by considering both past and future contexts. This resulting sequence becomes the input to the CNN layer, which extracts features from the previous sequences. The max pooling is applied to obtain a fixed-length vector and the softmax function is used by the output layer for input classification [130]. The results showed that the ensemble of BLSTM and CNN models led to an improvement in performance [134], [135], [136]. Another study by Zhang et al. [137] employed the BLSTM model for relation extraction. The architecture of BLSTM is shown in Figure 22.

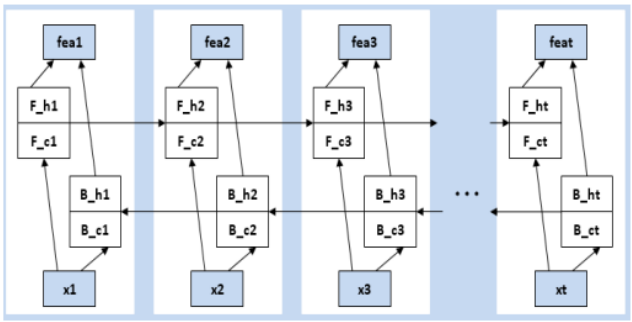
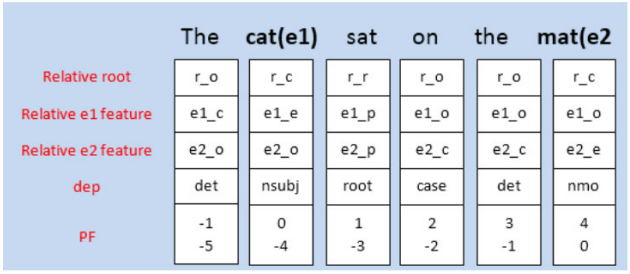


FIGURE 22. Bidirectional LSTM [137].

Given a sequence of data points, the network retains sequential and complete knowledge of previous and subsequent data points. The network finds appropriate features in the input sentence during feature extraction. Two categories of these features used to represent the relationships between two nouns include lexical features and relative position relationship features [137]. The authors [137] used the relative positions of words to extract features as illustrated in Figure 23.



	The	cat(e1)	sat	on	the	mat(e2
Relative root	r_o	r_c	r_r	r_o	r_o	r_c
Relative e1 feature	e1_c	e1_e	e1_p	e1_o	e1_o	e1_o
Relative e2 feature	e2_o	e2_o	e2_p	e2_c	e2_c	e2_e
dep	det	nsubj	root	case	det	nmo
PF	-1 -5	0 -4	1 -3	2 -2	3 -1	4 0

FIGURE 23. Example of relative position relationship features [137].

Position features are calculated using the relative distance between the current word and the two target nominal entities, e_1 and e_2. It is shown in the phrase (top of Figure 23) that the word "sat" has a relative distance of 1 from $cat(e_1)$ and -3 from $mat(e_2)$. As a basis for dependency features, Zhang et al. [137] used the Stanford dependency parser to capture long-distance relationships between these nominals. A feature embedding stage then maps every word to a real-valued vector, bringing together syntactic and semantic information. A vector representation of the original features is denoted by r^{kj} where j is the j^{th} type of feature. Given an embedding matrix $W^{wrd} \in \mathbb{R}^{d^w * |V|}$, where V is the size of the vocabulary, the embedding of each word is obtained using the matrix-vector product in Equation (32).

$$r^w = W^{wrd} * V^w \tag{32}$$

where V^w is a one-hot representation of one column of the two-dimensional array W^{wrd}. The initial feature embeddings of a given sentence $x = (w_1, w_2, \ldots\ldots w_n)$ are concatenated into $x_i = r_i^w, r_i^{k_1}, r_i^{k_2} \ldots\ldots\ldots\ldots r_i^{k_j}$, where r_i^w and $r_i^{k_j}$ are the embeddings of word x_i and the j^{th} feature type, respectively. The BLSTM is used in the third step of the process to create a sentence-level representation. From Figure 22 the outputs of the subnets for the i^{th} word are integrated as in Equation (33).

$$F_i = [F_{hi}, F_{ci}, B_{hi}, B_{hi}] \tag{33}$$

where F and B are forward and backward directions. The vector representation of two nominals is produced by concatenating the vector generated from the feature embedding and BLSTM layer as follows $[x_{e1}, F_{e1}, x_{e2}, F_{e2}]$. By using the nominals e_1 and e_2, matrix received from BLSTM is split into portions A, B, and C, as shown in Figure 24. The max pooling method is then used to extract the

vector from the A and B parts ($m1$), and B and C parts ($m2$), respectively. The sentence-level representation is created by combining $m1$ and $m2$.

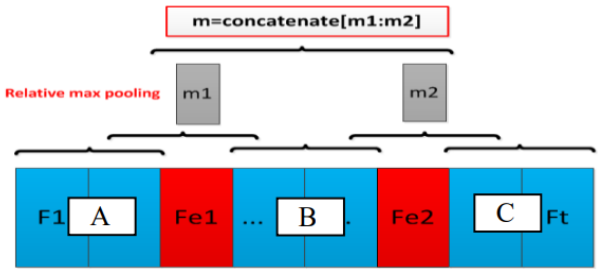


FIGURE 24. Sentence level feature vector construction [137].

The extracted feature vector produced by integrating the sentence-level features and lexical features using a multilayer perceptron becomes the input to a softmax classifier to predict semantic relationships. The authors concurred with the findings that BLSTM can identify long-distance relationships [137]. The experimental results revealed that BLSTM obtained an F1-score of 82.7% using two feature sets and 84.2% using multiple features.

The authors in [86], [138] implemented a two-step BERT fine-tuning model for relation extraction and classification using the DocRED dataset's document level. Relation extraction and classification are conducted in the first and second phases respectively. The second phase involves fine-tuning the BERT model to perform tasks such as token/ sequence classification or question answering [139]. In ontology construction, the attention masks are added to the token IDs to fine-tune the model for text classification. Gomes et al. [139] explained that BERT replaces 15% of word sequences with a masked token, then leverages contextual information from the surrounding, unmasked words within the sequence to predict the original values of the masked words. The technical process involves introducing a classification layer over the output generated by the encoder. The vector outputs are transformed into vocabulary dimensions by multiplying them with the embedding matrix and utilizing a softmax to compute word probabilities [139]. The BERT model demonstrated exceptional performance in word representation and comprehension of context-heavy texts.

A related study by Han and Wang [140] also employed the BERT model for relation extraction between entities in documents. The documents were fed into the pre-trained BERT model where the tokens of entities were averaged to produce the vector representations. The classification was done using a bilinear classifier and the results indicated a significant improvement in performance [141].

Liu et al. [142] presented the $BERT_{BASE}$ model in Figure 25, a Transformer with 12 layers. The model performs concept placement by identifying taxonomy (IS-A) relationships between a new concept and preexisting concepts. During the pre-training stage, the $BERT_{BASE}$ model took

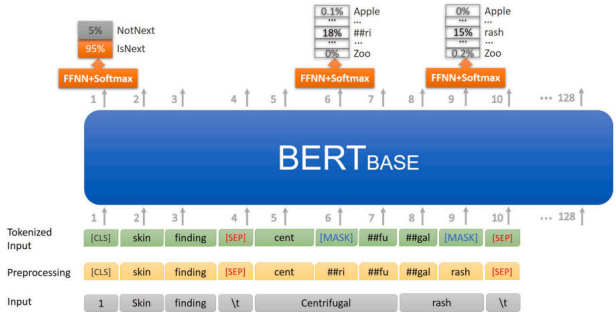


FIGURE 25. Pre-training the $BERT_{BASE}$ model [142].

preprocessed concept-based documents from the Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) as input and performed two training tasks: Masked Language Modelling (MLM) for prediction of masked words and Next Sentence Prediction (NSP) for learning relationships among sentences. The experiment extracted two sentences, ''Skin finding'' and ''Centrifugal rash'' from a sample document and pre-processed them. The $BERT_{BASE}$ model was trained to increase the probabilities of correct tokens ''$##ri$'' and ''$rash$'', as well as the classification label ''IsNext'', resulting in a new model encoded $BERT_{BASE+SNO}$, where the acronym SNO stands for SNOMED CT. For the fine-tuning stage, $BERT_{BASE}$ model was fine-tuned into an IS-A relationship classifier, and positive instances and negative instances were trained using existing IS-A relationships [143]. The model and classifier were trained simultaneously to predict IS-A links between ontology concepts. The parameters of $BERT_{BASE+SNO}$ and the classifier were fine-tuned to maximize the log probability of the correct labels. The error between predicted and true labels was backpropagated to improve the model's performance.

The study by Yelmen et al. [55] proposed a hybrid model of BERT and WordNet, to classify multi-class and imbalanced datasets. Initially, the WordNet terms were transformed into embeddings using the path2vec models. The use of BERT and WordNet was either external or internal, as illustrated in Figure 26. The internal combination of BERT and WordNet entails integrating the representation generated by one model into the framework of the other. As shown at the bottom of Figure 26, both BERT and WordNet generate word embeddings of n dimensions from a given text with n tokens. The vertical concatenation method produces a fine-tuned embedding by feeding it into the BERT network. For instance, given the sentence ''This is a nice apple'' for a classification task. The BERT model calculates the embeddings of the subsequent tokens [CLS], $this$, is, a, $nice$, $apple$, [SEP] as illustrated on the left of Figure 27. The WordNet embeddings for 'be' (lemmatized as 'is'), 'nice,' and 'apple' are integrated into the word embeddings matrix (right of Figure 27), since BERT recognizes these embeddings as sentence constituents [144], [145]. However, the results showed that combining these embeddings with BERT did

not improve the model's accuracy on sentence similarity and natural language inference compared to the BERT-only model. The BERT model calculates the embeddings of the subsequent tokens [*CLS*], *this*, *is*, *a*, *nice*, *apple*, [*SEP*] as illustrated on the left of Figure 27. The WordNet embeddings for 'be' (lemmatized as 'is'), 'nice,' and 'apple' are integrated into the word embeddings matrix (right of Figure 27), since BERT recognizes these embeddings as sentence constituents [144], [145]. However, the results showed that combining these embeddings with BERT did not improve the model's accuracy on sentence similarity and natural language inference compared to the BERT-only model.

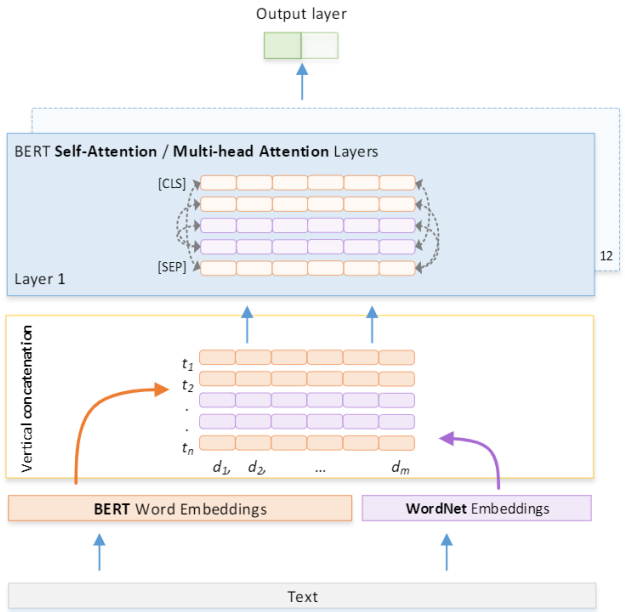


FIGURE 26. Internal Inclusion in combining, BERT-WordNet. [144].

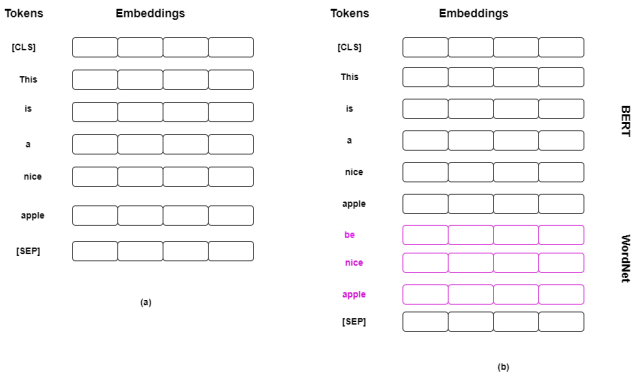


FIGURE 27. Vertical Concatenation of BERT-WordNet. [144].

Lu et al. [146] proposed a hybrid model, namely, VGCN-BERT, which integrates the Vocabulary Graph Convolution Network (VGCN) and BERT model for text classification. The integration was necessitated by the fact that the BERT model fails to capture global information about the vocabulary of a language. Therefore, as a countermeasure, Lu et al. [146] integrated VGCN and BERT where the strengths of both models are combined. The network encodes global language information and interacts with the word embedding and graph embedding through a self-attention encoder in BERT, thus integrating local and global information in the final representation [147]. The VGCN-BERT model is depicted in Figure 28.

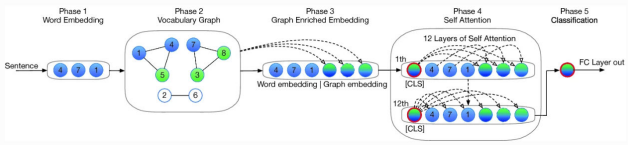


FIGURE 28. Hybrid model, VGCN-BERT. [146], [147].

In Figure 28, the input sentence embeddings are combined with vocabulary graph to create a graph embedding, only relevant input is extracted and embedded. The Self-attention layers are applied to the integrated representation, and the final embedding is passed into a fully connected layer for classification. Mathematically, VGCN and BERT can be integrated as follows: given a graph of nodes (P) and edges (E), $G = (P, E)$ with $|P| = n$, for a single layer in a Graph Convolution Network (GCN), one have the representation in Equation (34) [147].

$$H = \widetilde{A}XW \tag{34}$$

where $X \in \mathbb{R}^{n*m}$ is a two-dimensional array input of n nodes and m dimensions of features, $W \in \mathbb{R}^{m*h}$ the two-dimensional array of weights, $\widetilde{A} = D^{-1/2}AD^{-1/2}$ a symmetric adjacency matrix which is normalized. Related words are then convolved, by constructing a GCN on the vocabulary rather than the document using a convolution layer defined in Equation (35) [147].

$$h = (\widetilde{A}x^T)^T W = x\widetilde{A}W \tag{35}$$

where $\widetilde{A}^T = \widetilde{A}$ is the graph of the vocabulary, $x\widetilde{A}$ is the part of the graph related to the sentence x, W the weights matrix of the hidden state vector for a document of $|V| * h$ dimension. Given m documents, the single layer of GCN is represented in Equation (36) [147].

$$H = X\widetilde{A}W \tag{36}$$

The authors in [148] and [149] argued that a two-layer GCN outperforms a one-layer GCN, with more layers providing minimal improvement. The two-layer GCN is computed with Equation (37) [147].

$$VGCN = ReLU(X_{mv}\widetilde{A}_{vv}W_{vh})W_{hc} \tag{37}$$

where m is the batch size, v the size of vocabulary, h the size of the hidden layer, and c the size of class or sentence embedding. The equation creates a convolution layer of the graph, which represents part of the graph suitable to the

input using $X_{mv}\widetilde{A}_{vv}$, then executes the two-layer convolution by integrating the input sentence with related words in the vocabulary graph. For text classification, BERT comprises a word embedding module, a transformer module using multi-layer multi-head self-attention stacking, and a fully connected layer using output sentence embedding. The self-attention algorithm computes the attention score by utilizing a query Q with a pair constituted of a key K and value V as in Equation (38) [147].

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \tag{38}$$

where d_k represents the size of the query and the associated key vectors. The square root of d_k is a scaling factor for controlling the scale of the attention.The VGCN-BERT model incorporates both vocabulary graph embedding and sentence sequence, which helps maintain the order of words and fully employ background information. Local and global embeddings are integrated into a 12-layer and 12-head self-attention encoder, resulting in a formulated VGCN in Equation (39) [147].

$$G_{embedding} = ReLU(X_{mev}\widetilde{A}_{vv}W_{vh})W_{hg} \tag{39}$$

where $ReLU$ is an activation function, W_{hg} is the output of graph embedding with size g, m the size of the mini-batch, and e the dimension of the word embedding. The experimental results showed that the VGCN-BERT model performed better compared to the VGCN and BERT when used separately.

Xue et al. [150] proposed the VGCN-BERT-BiLSTM model for text classification. The model ensembles the vocabulary graph convolutional network (VGCN), the BERT model, and the Bi-directional Long-Short Term Memory (BiLSTM) model as shown in Figure 29, to perform text classification. The vocabulary and the dependency graph are created and subsequently employed to train a two-layer Graph Convolutional Network (GCN). The word embedding from the BERT model becomes an input to the GCN, including the two hidden outputs. The word vector is then combined with the feature vector of the pre-trained BERT and passed to the first layer of the BiLSTM model to extract the semantic dependency information between words within a single sentence [150]. The assignment of word weights is done in the attention layer. The last layer of BiLSTM sums the word weights to form a sentence representation.

A number of studies used BERT for ontology alignment [151], [152], [153], [154]. The authors in [153], [154], and [154] proposed a BERTMap for ontology alignment. BERTSubs predict the absence of concept subsumptions within an OWL ontology and determine the subsumptions between concepts derived from two separate OWL ontologies to facilitate alignment. Neutel and de Boer [151] concluded that BERT performed well for the automatic alignment of two occupational ontologies. Rudwan and Fonou-Dombeu [152] proposed an ontology alignment method that integrates

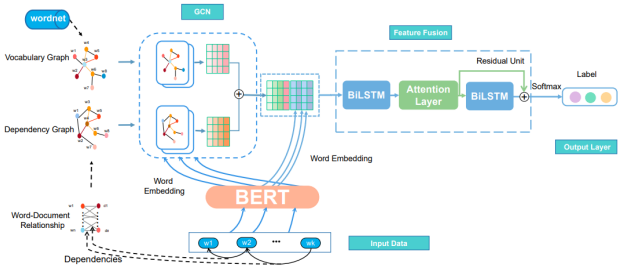


FIGURE 29. Hybrid model, VGCN-BERT-BiLSTM [150].

fuzzy string-matching algorithms and the BERT model for conducting an ontology alignment, the BERT model showed promising results.

A study by Gutierrez et al. [155] employed GPT-3 in context learning for biomedical information extraction. The experiment was compared with BERT on two representative biomedical information tasks. The experiment started by reformulation of NER and RE as language generation tasks followed by transformations of every example into prompts by adding subject and object entities to the prompt. The output was then calibrated using linear transformation to equalizes all label probabilities generated by GPT-3 [155]. KNN was then employed to identify the most similar example in the training set as the few-shot in-context prompt for each text example. The results indicated that GPT-3 underperforms compared to BERT model as shown by the small gains in accuracy even when more data was available [155]. A similar study by Bellan et al. [156] also employed GPT-3 for extracting business process entities and relations from text. The results indicated that GPT-3 was able to extract activities, participant and perform relation between a participant and an activity.

Ouyang et al. [27] proposed the InstructGPT model that was built on GPT-3 [27], which uses human feedback through a technique called ‘‘reinforcement learning from human feedback’’ (RLHF) to make sure it follows specific guidelines. The authors built a dataset of rankings of model outputs which was used for further fine-tuning the supervised model using reinforcement learning from human feedback to produce the InstructGPT model which consists of 1.3 billion parameters. This approach led to ChatGPT, which became popular for its ability to hold helpful conversations on many domains. The results indicated that InstructGPT shows some improvements in truthfulness and reduction of toxic output, however, it makes simple mistakes for example assuming the premise is true when presented with an instruction whose premise is false. Another study by Agrawal et al. [157] demonstrated the effectiveness of InstructGPT model in extraction of clinical information with minimal training data. Although the model was not specifically trained for clinical tasks, it performs well in zero- and few-shot scenarios. The study introduced new benchmark datasets originated from the CASI dataset to address issues of dataset shifts and lack of public clinical corpora. Results showed that GPT-3 systems

outperform existing baselines in various clinical information extraction tasks, including span identification, token-level sequence classification, and relation extraction.

Groza et al. [158] conducted a study using GPT-4 in-context learning for concept recognition. The experiment used ontology concepts from Human Phenotype Ontology together with phenotype concept recognition task to produce patient profiles. The produced results showed a good F1-score measure surpassing the state of the art performance however,GPT-4 model suffers from unpredictable results, steep expenses, and inconsistency across multiple attempts with the same prompt and input pose significant challenges when employing these Large Language Models for this specific task.

Qiu and Jin [159] investigated the effectiveness of ChatGPT and fine-tuned BERT for domain-specific sentence classification, extraction and mapping. While both models achieved comparable performance, fine-tuned BERT struggled with brevity. Interestingly, ChatGPT exhibited significant improvement in a few-shot learning scenario. Additionally, it leveraged domain knowledge to filter irrelevant data, enhancing dataset quality. For content generation, ChatGPT, in a zero-shot setting, produced informative and coherent responses to domain-specific questions. However, it occasionally included extraneous details that could burden readers. Nonetheless, ChatGPT holds promise for tasks like data labeling, knowledge transfer, and eliciting domain-specific knowledge. Its ability to function with minimal guidance suggests it has the potential to significantly improve domain experts' efficiency.

VII. AXIOM LEARNING

Axiom learning involves the extraction of axioms and rules, and generating predictions [7]. Petrucci et al. [160] proposed a Recurrent Neural Network (RNN) based architecture for axiom extraction from text. The aim was to convert natural language knowledge into a logic-based specification using an RNN model. The process is in two phases: sentence transduction and sentence tagging. Sentence transduction transforms a sentence in natural language into a series of logical symbols called a formula. Sentence tagging identifies and classifies words within a sentence according to their respective roles, which may include representing a concept, a role, a number, or a general term as shown in Figure 30.

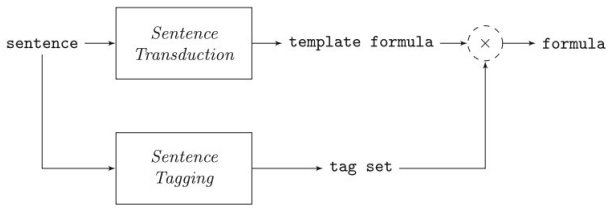


FIGURE 30. Axiom Extraction Pipeline [160].

The following natural language sentence, "A goat is a mammal that has four legs and produces milk," can be converted to logical symbols to produce: $Goat \sqsubseteq Mammal \sqcap = 4\ have.Leg \sqcap \exists produces.Milk$. The sentence tagging for the same sentence produces the following: $C_0 \sqsubseteq C_1 \sqcap = N_0 R_0.C_2 \sqcap R_1.C_3$ where C, R, N are used for concepts, roles, and numbers, respectively, with subscripts indicating their order. The outputs of the two phases are combined to produce a formula. When used for sentence tagging, RNNs accept embedding vectors as input into the network, then a word window of width w is created and consists of a short sequence of words centered on the i^{th} word in a sentence as shown in Figure 31. The vectors in the window are then concatenated and passed to the recurrent layer, which produces a vector of the same size as the possible tag count. Each component of this vector represents a score for a specific tag. At time step t, the network predicts the appropriate tag for the current input word.

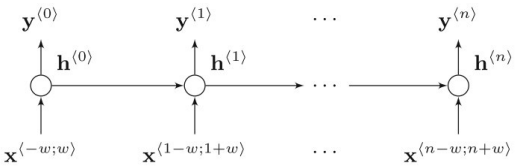


FIGURE 31. RNN for Sentence Tagging [160].

A probability distribution across all possible tags is modeled by applying a softmax over the scores. The tag assigned to the k^{th} word is the most probable, which is the argmax over the output vector $y^{<t>}$.

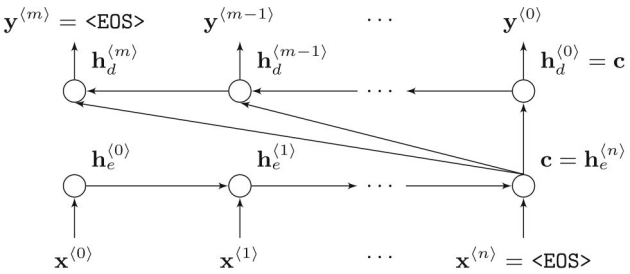


FIGURE 32. RNN for Sentence Transduction [160].

When it comes to sentence transduction, a double-layered recurrent network called the Recurrent Encoder-Decoder (RED) is set up, as shown in Figure 32. Here, the first RNN encodes the input sequence and generates a vector c that captures the comprehensive representation of the entire sentence at its final time step. The vector is employed as a input for a second RNN which decodes the information from this vector into a new sequence. The technique ensures a semantic connection between the two sequences due to the distributed representation of the input sequence in each decoding step. However, there is no direct one-to-one mapping between the input and output symbols. Figure 32 shows the transduction without the feedback of the output from the previous step, resulting in a simpler model Although

the model was trained with limited data, the experiment demonstrated the potential of Deep Learning for axiom extraction.

Another study by Peng et al. [161] demonstrated the use of a Neural Reasoner, a framework for neural network-based reasoning over natural language sentences for extraction of rules and axioms. The Neural Reasoner utilizes a unique interaction-pooling mechanism to analyze multiple facts and a deep architecture to model intricate logical relationships in reasoning tasks. Recurrent neural networks were employed to convert questions and facts to vector representation in the encoding layer. These are then updated in the reasoning layers using deep neural networks and pooling. Figure 33 shows the framework that uses a Neural Reasoner to perform axiom learning. The framework comprises the encoding layer (bottom of Figure 33), reasoning layer-1 to reasoning layer-L (middle of Figure 33), and the answering layer (top of Figure 33). Recurrent neural networks (RNNs) are employed to convert questions and facts into vector representations at the encoding layer and results are sent to the first reasoning layer. In each reasoning layer, a deep neural network (DNN) models the interaction between a question representation $q^{(l-1)}$ and a truth representation $f_k^{(l-1)}$, resulting in a new reality representation f_k^l. The global updated representation q^l is combined with individual updated fact representations $(q_1^l, q_2^l, \ldots, q_k^l)$ through a pooling operation. In Layer-L, the interaction net (DNN_L) returns only the question update, summarized by the pooling operation, and acts as an input to the Answering Layer.

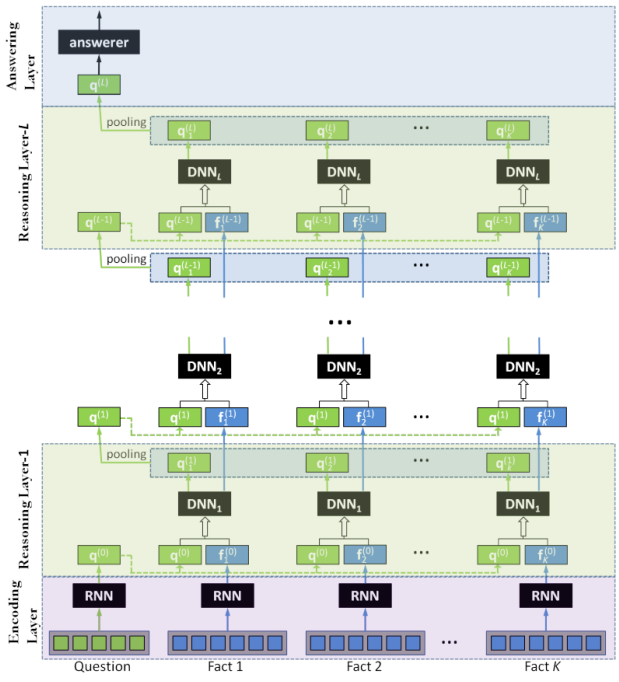


FIGURE 33. A framework for Neural Reasoner with L reasoning layers, operating on one question and K facts. [161].

Cai et al. [162] proposed a symbolic reasoning method using deep neural networks (DNN) for axiom discovery. The method uses symbolic manipulation, combining the learning ability of deep neural networks with the reasoning ability of a logical system. The logical system generates training examples in order for deep neural networks to learn, score, and extract axioms. The results indicated that more hidden layers improve symbolic manipulation learning and axiom discovery, enabling deep neural networks to learn important logical axioms.

VIII. EVALUATION OF DEEP LEARNING MODELS FOR ONTOLOGY CONSTRUCTION

Precision, Recall, and F1-score are key metrics for evaluating deep learning models in ontology learning [176]. Precision measures the "true positive rate," telling you what portion of the model's positive predictions are actually correct. Recall, on the other hand, calculates the "completeness" by capturing how many true positives the model actually identifies. F1-score is the harmonic mean of Precision and Recall, which serves as a comprehensive measure and the preferred choice for model comparisons.

IX. ONTOLOGY CONSTRUCTION SYSTEMS

Building on the idea of capturing contextual meaning, He et al., [153] developed a BERTMap-based system for ontology matching. The system extracts text from ontologies, generates contextual embeddings using BERT, and predicts semantic mappings between concepts. Moreover, it uses the inherent structure and logic of ontologies to refine the alignment, leading to reportedly higher F1 scores. In research by Sousa and Couto [103], the BiOnt system used deep learning to automatically extract relationships from four key biomedical ontologies: gene ontology, human phenotype ontology, human disease ontology, and chemical entities of biological interest. This demonstrates the potential of deep learning for extracting valuable information from diverse ontologies in the biomedical domain. The system was tested with three datasets representing three different ratios of biomedical units. BiOnt improved the F1-score by 4.93% in drug-drug interactions (DDI corpus), by 4.99% in phenotype-gene associations (PGR corpus), and by 2.21% in chemical-induced [103]. BioBERT, a domain-specific language representation model pre-trained on large-scale biomedical corpora, was proposed in [163] for biomedical named entity recognition, biomedical relation extraction, and biomedical question answering. Results showed that BioBERT outperforms BERT on those three tasks [163].

Ganesan et al. [96] proposed a system for the automatic population of a Person ontology graph from unstructured data using the RNN and the BRNN models for entity classification and relation extraction. A new dataset containing 36 Personal Data Entity Types and 9 Personal Data Entity Relations was employed. The approach improved the state-of-the-art methods for fine-grained entity classification using lightweight features. However, the model may not be trained on different domains or structured data, posing a weakness that requires further research. Ayadi et al. [35]

proposed a deep learning-based NLP ontology population system to populate the biomolecular network Ontology. The authors jointly exploit deep learning and natural language processing techniques to identify, extract, and classify new instances from textual data. The results attested that the proposed method was efficient for the ontology population. Another study by Alam et al. [164] proposed a fully automated framework for extracting concepts and building Knowledge Graphs for different clinical domains. The framework employed RNN coupled with BioBERT for concept extraction and relation discovery. The proposed framework obtained better results than baseline models. In another study, Fang et al. [90] designed a drought disaster ontology through concept identification and relation extraction. The study ensembled the BERT, BiLSTM, and CRF deep learning models [91], [93]. The BERT-BiLSTM-CRF presented good results determining drought disaster named entities with high precision, recall, and F1-score [93].

Chandolikar et al. [165] demonstrated how deep learning models outperform the conventional machine learning models in designing a science domain ontology using the Marathi language. The experiments were performed on science keywords to build an ontology in the Marathi language for a knowledge search engine that provides correct information about science terms in the Marathi language to school-going children [165].

A. DATASETS USED IN ONTOLOGY CONSTRUCTION

The efficacy of deep learning models for concept extraction, relation extraction and classification, axiom learning and ontology alignment, is dependent on the careful selection of appropriate datasets. Table 3 shows the commonly used datasets for ontology construction tasks using deep learning models. The datasets in Table 3 are open and publicly available for use. For each dataset, the task it is used for, its size in terms of number of sentences/entities and sources are given.

X. COMPARATIVE ANALYSIS AND DISCUSSION OF DEEP LEARNING MODELS

Several Deep learning models such as CBow, Skip-gram, BERT, RNN, LSTM, and BiLSTM as well as their combination were reviewed in this paper showing how the models were applied for term extraction, relation discovery, and axiom learning [12], [36], [36], [37], [38], [39], [86], [169]. The comparative description of the Deep Learning model reviewed is provided in Table 4.

Deep learning techniques for word embedding exhibited efficient performance in semantic similarity. However, the Word2vec model, which employs the CBow and Skip-gram algorithms presents a challenge in terms of providing an exact formal description available in an ontology [151]. The BERT model, serving as context embedding strikes a balance between syntactic and surface features and proved to be more effective in identifying context [25] resulting in improving word representation. The review also highlighted the significance of unsupervised pre-training of word vectors as a crucial component in deep learning for term extraction and relation discovery in ontology learning [142].

Table 4 shows several deep learning models that have been applied for word embeddings. These models rely on attention mechanisms. For example, the BERT directs its attention to nearby and consecutive series of words, resulting in the acquisition of localized context information [25]. Also, the BERT model positions each word within its immediate context to produce a context-specific representation. Although the BERT model proved effective for many natural language processing tasks, it does not capture information that covers the entire spectrum of a language. The BERT's representations often lack long-range connections and may ignore the broader context within a text [146].

Researchers have sought innovative solutions by proposing the ensembling of multiple models to complement each other's weaknesses. The insights gathered from the review showed a promising approach: combining BERT with the Graph Convolutional Network (GCN) for word embeddings. The integration proved beneficial because vocabulary graphs, as implemented in GCN, have the potential to offer additional layers of understanding and context [56], [58]. The construction of a vocabulary graph for a specific collection of documents relevant to a particular field holds the key to unlocking the benefits of this approach. The GCN facilitates the capture of general dependencies obtained in pre-trained models and empowers the modeling of application-specific dependencies, bridging the gap between localized and extensive context awareness in language processing.

Researchers have explored various deep-learning models for term extraction including LSTM, RNN, CNN, GRU, DBN, and a hybrid of the BERT-BiLSTM-CRF model. Among this array of models, the Gated Recurrent Unit (GRU) was observed to outperform its counterparts, showcasing its performance in this particular domain [72]. The distinguishing factor of GRU is its capacity to process textual data effectively and capture and comprehend long-term dependencies within the text. Another observation emerges when considering the impact of combining the GRU model with input encodings. This synergistic fusion obtains an improvement in the performance of the GRU model. A critical turning point in maximizing the capabilities of the GRU model is merging the model with a domain-specific embedding corpus. Experiments reported in [72] and [78] demonstrated that this integration sets the GRU model to greater heights of performance. Combining a domain-specific embedding corpus and the GRU model allows the model to comprehend and contextualize terminology within the specific domain to produce impressive term extraction precision and recall. The GRU model combined with domain-specific embeddings represents a potent and promising approach for term extraction within the context of deep learning research and applications.

The RNN, CNN, LSTM, GCN, BiLSTM, GRU, GPT and BERT models have emerged as highly promising approaches

TABLE 3. Commonly used datasets for ontology construction tasks using deep learning models.

Dataset	Associated task	Sentences/Entities	Sources
SNOMED CT	concept placement, relation extraction		[142]
PubMed	relation extraction	4.5billion	[163]
SemEval-2010 task 8	relation extraction	21046	[129] [44] , [137] [122] [101]
MEDLINE/PubMed	ontology learning	14,056,762	[14]
Open Access Series of Imaging Studies (OASIS)	ontology construction	416	[112]
BiodivNER	relation extraction	2,398	[166]
SST-1 Stanford Sentiment Treebank	classification	11 855	[167] , [8]
MR	classification		[167]
SemEval 2013: Task 9	DDI Extraction Corpus relation extraction	5 028	[103] [122] [101]
BioRel	relation extraction	69,513	[168]
Adverse Drug Effect (ADE)	relation extraction	5,063	[168]
NYT10	classification	112,941	[128]
SQuAD	word embeddings	100 000	[45]
BooksCorpus dataset	text representation		[47]
TACRED	relation extraction, relation classification		[44]

to relation extraction and classification. This review revealed significant insights into their strengths and limitations. RNNs were able to predict future words based on past words, highlighting their potential in capturing sequential information in text [8], [8], [129]. However, RNN models was limited in capturing long-term dependencies effectively. CNNs, on the other hand, were very good at extracting higher-level features invariant to local translations in the context of ontology relation extraction [110]. However, they fail to capture long-term dependencies due to the inherent locality of convolution and pooling operations [107]. LSTMs are limited in identifying distant relations [137]. In trying to resolve this limitation the Bidirectional Long Short-Term Memory network (BiLSTM) model was introduced. The BiLSTMs, with their bidirectional nature, have demonstrated remarkable capabilities in identifying long-distance relations among terms, improving the use of LSTM-based models. BERT model performed exceptionally well [142] in learning the underlying attention of the input text and each term's relationships to its context, using its pre-training and fine-tuning stages [143]. Although the BERT model was able to comprehend context-heavy texts, the model struggles with higher-level abstraction and inference tasks due to a lack of explicit semantic knowledge, thus necessitating inference for unspecified relationships [55], [144]. GCNs were successful in capturing global information about the vocabulary of a language however their limitation is that they are not able to capture local information when performing text classification, because of their sparse adjacency matrix and, therefore, fail to fully exploit context-dependent information [146]. Furthermore, GCNs cannot handle unseen documents because of its transductive learning mode [146].

To Address some of the abovementioned limitations, researchers have embarked on novel research of hybridizing these models to complement each other's weaknesses. GCN was ensembled with the BERT model [55], [144] were BERT captures the micro-level details of language by processing individual words, while GCN focuses on the macro-level by analyzing relationships between words. This complementary perspective on meaning proves powerful when both models are combined, as seen in improved performance [146], [147]. The introduction of a model that ensemble CNN and RNN for relation discovery yields better results compared to when the models were applied separately [8]. Furthermore, the BERT-BiLSTM-CRF model, which combines BERT, Bidirectional LSTM (BiLSTM), and Conditional Random Fields (CRF), showed promising results in concept extraction and relation discovery, surpassing the performance of standalone models. Also, GCN, BERT, and BiLSTM were ensembled for relation discovery and produced improved results [150]. The BERT model was integrated with vocabulary graphs and WordNet for relation discovery to uncover abstracted relationships [55], [144]. This fusion of BERT with a vocabulary graphs enhanced the fusion of local information represented by BERT with global vocabulary information. Consequently, this facilitated information aggregation throughout the layers of the attention mechanism [144], [146].

The review provides irresistible evidence that ontology construction systems combined with domain-specific ontologies or knowledge bases deliver notable improvements in the performance of deep learning models for their respective tasks [72], [73], [90]. The improvement is certified by the remarkable performance of the BiOnto system which outperforms other prominent deep learning models like BERT, BioBERT, and BO-LSTM [96], [98], [103]. However, despite the evident prowess of these systems within their respective domains, a crucial limitation emerged in their

TABLE 4. Deep Learning Models Reviewed and Ontology Learning Tasks.

	Word Embedding	Term Extraction	Relation Discovery	Axiom Learning	Ref
BERT	✓		✓		[25] [55] [55] [44] [86] [138]
GRU	✓	✓	✓	✓	[72] [78] [73]
CBow	✓		✓		[37] [96] [36] [38] [12] [14]
Skip-Gram	✓		✓		[36] [37] [38] [12]
CNN		✓	✓	✓	[107] [110] [111] [112] [108] [171]
GCN	✓		✓		[148] [149]
RNN	✓	✓	✓	✓	[129] [122] [160]
LSTM (Elmo)	✓	✓	✓		[72] [78] [73]
MIMLCNN	✓		✓		[128]
BiLSTM	✓		✓		[137] [122]
VGCN-BERT				✓	[146] [147]
BERT-BiLSTM-CRF		✓	✓		[79] [91] [92] [93] [94]
VGCN-BERT-BiLSTM			✓		[150]
G-BERT	✓		✓		[56]

lack of generalizability across different domains. The models do not easily adapt to tasks or knowledge domains outside their predefined scope. Also if a system was trained using unstructured data, it may face significant constraints when attempting to process structured data. The adaptability and usability of such systems may be hindered when presented with data of varying structures and formats [96].

In light of the above discussion, Deep learning models reviewed have proved to be efficient in processing unstructured data, learning prior knowledge from input data, supporting different learning paradigms, extracting features from labeled or unlabelled data, and being efficient in discovering relationships and patterns [10], [73].

XI. CHALLENGES AND FUTURE DIRECTIONS OF RESEARCH

The adoption of deep learning in ontology construction has yielded numerous advantages. Despite the widespread interest generated by the breadth of this research domain, several unresolved challenges persist. These challenges are examined, and potential future research avenues with fresh perspectives are suggested to foster the continued advancement of this field.

A. SCARCITY OF LABELLED DATA AND UNAVAILABILITY OF VOLUMINOUS DATA

Deep learning relies on extensive datasets to achieve optimal performance. When datasets are insufficient in size, applications may not fully harness the potential of deep learning techniques.This has been witnessed by the application of transformer models such as BERT [139], [140], [141], [142] and GPT [157], [158], which have shown remarkable performance in natural language understanding and generation tasks on large datasets. Adapting these models for ontology construction will produce efficient and accurate representation learning of ontology concepts and

relationships. Adding on to the transformer models are the Unsupervised learning techniques, such as autoencoders and generative adversarial networks (GANs), which have shown promising avenue for the construction of ontology without requiring labeled data. Unsupervised learning approaches can use large amounts of unlabelled data to automatically discover underlying patterns and structures within ontologies. Therefore, facilitating the creation of ontologies from heterogeneous data sources and support the integration of diverse knowledge representations. However, it has been highlighted in this review that GANs suffers from high computational cost and memory requirements [20], [119] and it had been suggested that transfer learning can address the issue however transfer learning has not been exhausted. Transfer learning within an ontology construction, allows for the reuse of deep learning models initially designed to address specific challenges. This reutilization becomes particularly advantageous when applied to another task within the same architecture, especially when data scarcity is a concern for the secondary task. In the domain where there is no voluminous data, the concept of transfer learning for deep learning can be applied to minimize lack of large data size issues. Therefore, we suggest applications of the concept of transfer learning in ontology construction.

B. DEEP LEARNING FOR RULES AND AXIOMS LEARNING

The review has revealed that there is little effort done in extracting rules and axioms using deep learning models for ontology construction [171], [172]. This was due to the fact that deep learning models learn patterns in raw data, therefore, representing symbolic knowledge, such as rules and logical constraints, in a format suitable for deep learning can be non-trivial [171]. Further research should be done to develop novel approaches for automatically extracting, inferring, and codifying rules and axioms within ontologies based on progress made in deep learning techniques.

C. DEEP LEARNING FOR ALIGNMENT AND INTEGRATION OF EXTRACTED ONTOLOGY ELEMENTS

The review has addressed the use of deep learning to extract ontology's concepts, relations and axioms from unstructured textual contents as well as from existing ontologies in some cases. Further research needs to investigate suitable deep learning structures to align and merge/integrate these ontology's constituents to build complete ontologies.

D. DEEP LEARNING METHODS FOR AUTOMATION OF QUALITY ASSURANCE AND EVALUATION PROCESSES IN ONTOLOGY CONSTRUCTION

Some evaluation metrics require gold standard datasets or reference ontologies for comparison. However, obtaining such datasets can be labor-intensive and may not always be feasible, particularly for specialized domains. Also, limited incorporation of user feedback and scalability issues may affect the quality assurance and evaluation processes. Quality assurance and evaluation processes in ontology construction can be automated using deep learning methods through learning how to detect errors, inconsistencies, and redundancies within ontologies. We suggest use of Neural network-based anomaly detection techniques [173], abnormal patterns or outliers [174] in ontology structures can be identified.

E. HYBRID AND ENSEMBLE OF DEEP LEARNING MODELS FOR ONTOLOGY CONSTRUCTION TASKS

As illustrated in Table 4, the integration of hybrid deep learning models in ontology construction tasks has received comparatively less attention from the research community. However, it's believed that hybrid deep learning models outperform individual counterparts by leveraging the strengths of respective models while mitigating their respective limitations. Additionally, ensemble deep learning models remain unexploited in the context of ontology construction. There is potential value in exploring the application of hybrid and ensemble deep learning approaches to tackle complex challenges within ontology construction. We suggest researchers to dig into the concepts of hybrid and ensemble deep learning models in ontology construction.

F. OPTIMIZATION OF DEEP LEARNING MODELS

The review also discovered that research focused on the usefulness of deep learning models for feature engineering rather than optimizing the model to improve its effectiveness. That is another direction that needs exploration. Also the deep learning models rely on manual adjustment of hyper-parameters. This manual process is burdensome, time-consuming, and lacks a standardized systematic approach for determining optimal hyper-parameter values. It is recommended that future efforts explore the utilization of global optimization algorithms, such as cuckoo search, firefly, artificial bee colony, and immune system algorithms, to optimize the hyper-parameters of deep learning models for ontology construction tasks. This suggestion is supported by evidence presented in [175], demonstrating that deep learning algorithms optimized through global optimization algorithms outperform conventional counterparts where hyper-parameters are optimized manually.

G. DIFFICULTY IN CHOOSING AN APPROPRIATE DEEP LEARNING MODEL

Various deep learning algorithms and their variations exist, presenting researchers and developers with a challenge in determining the optimal choice for ontology construction tasks. Choosing the right deep learning model for term extraction, relation extraction, or classification in ontology construction requires careful consideration of task requirements [10], data availability, interpretability, and domain expertise. It often involves a combination of experimentation, domain knowledge, and methodological expertise to identify the most effective approach for a given ontology construction task. Given the increasing prominence of deep learning in ontology construction tasks, there is a pressing need for insights into the performance of different deep learning

models across tasks. Such insights would greatly facilitate the selection of the most suitable deep learning algorithm for specific ontology construction tasks.

XII. CONCLUSION

A thorough examination of the utilization of deep learning models in ontology construction has been undertaken. Major deep learning architectures, including members of the GPT family, BERT transformers, RNN, LSTM, CNN, and DBN, were scrutinized for their applications across various ontology construction tasks, such as concept extraction, relation extraction and classification, axiom learning, and ontology alignment and merging. Various deep learning models have been deployed to address diverse challenges in ontology construction. According to the reviewed literature, the BERT model received a lot of attention across the ontology learning stages due to its ability to consider contextual information, followed closely by GPT. The trajectory of publications suggests a burgeoning interest in the adoption of deep learning for ontology construction tasks, which is anticipated to persist due to evolving trends and emerging research opportunities. Additionally, unresolved research challenges and novel perspectives for future research directions were deliberated upon to offer guidance for tackling identified obstacles. This review serves as valuable introductory material for new researchers in the field of ontology construction and provides a benchmark for proposing innovative deep learning approaches to address ontology construction challenges by seasoned experts.

Deep learning models show promising potential for ontology learning when employed as ensemble models. Combining different models allows them to effectively handle the diverse tasks involved in ontology construction, from concept extraction to relation identification. In particular, it was found that the BERT model was more successful. The other main finding of the review was that, when a domain-specific ontology is combined with the deep learning models for word embeddings, term extraction, and relation discovery, the models performed better. The shift towards Deep Learning in word embedding, concept and relation classification, and axiom learning is proof of the advancement of ontology learning/construction research. Researchers are going ahead, utilizing the capabilities of Deep Learning to unlock the complexities of language and text, with the ultimate goal of advancing the capacity to understand and interpret the wealth of information encapsulated within textual data and facilitate automated ontology construction.

The review emphasizes how important it is to investigate novel ontology construction strategies, such as deep learning methods. These algorithms can reduce manual effort and enable the automation of certain parts of the ontology building process. Subtle nuances and semantic relationships that may be difficult for standard approaches to capture can be captured by deep learning models. However, there are a number of obstacles that must be addressed, including the requirement for large training datasets, which may be scarce in certain domains. Also it is important to gain trust and understanding among domain experts because it ensures interpretability and explainability of deep learning models. Additionally, deep learning approaches introduce new complexities related to model training, optimization, and evaluation, that requires interdisciplinary collaborations between computer scientists, domain experts, and ontologists. While deep learning isn't without its hurdles, the potential benefits are undeniable. As the mountains of big data continue to grow, deep learning will likely become even more crucial in extracting valuable insights and guiding decision-making across various fields. Deep learning poses great promise for transforming ontology construction practices, though due deligence by researchers and practitioners should be exercised when navigating the opportunities and challenges highlighted in this review. Harnessing the strengths of deep learning while addressing its limitations, ontology construction can advance towards more efficient, accurate, and scalable ontology construction methodologies, resulting in enhancing knowledge representation and semantic interoperability across various domains.

REFERENCES

[1] A. C. Khadir, H. Aliane, and A. Guessoum, "Ontology learning: Grand tour and challenges," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100339.

[2] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 1–36, Aug. 2012.

[3] A. Konys, "Knowledge repository of ontology learning tools from text," *Proc. Comput. Sci.*, vol. 159, pp. 1614–1628, Jan. 2019.

[4] J. Wątróbski, "Ontology learning methods from text—An extensive knowledge-based approach," *Proc. Comput. Sci.*, vol. 176, pp. 3356–3368, Jan. 2020.

[5] A. Browarnik and O. Maimon, "Ontology learning from text: Why the ontology learning layer cake is not viable," *Int. J. Signs Semiotic Syst.*, vol. 4, no. 2, pp. 1–14, Jul. 2015.

[6] J. L. Ochoa, M. L. Hernández-Alcaraz, R. Valencia-García, and R. Martínez-Béjar, "A semantic role-based approach for ontology learning from Spanish texts," in *Proc. Int. Symp. Distrib. Comput. Artif. Intell.* Berlin, Germany: Springer, 2011, pp. 273–280.

[7] J. L. Ochoa, R. Valencia-García, A. Perez-Soltero, and M. Barceló-Valenzuela, "A semantic role labelling-based framework for learning ontologies from Spanish documents," *Expert Syst. Appl.*, vol. 40, no. 6, pp. 2058–2068, May 2013.

[8] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.

[9] P. Buitelaar, P. Cimiano, and B. Magnini, "Ontology learning from text: An overview," in *Ontology Learning From Text: Methods, Evaluation and Applications*, vol. 123. The Netherlands: IOS Press, 2005.

[10] F. N. Al-Aswadi, H. Y. Chan, and K. H. Gan, "Automatic ontology construction from text: A review from shallow to deep learning trend," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 3901–3928, Aug. 2020.

[11] A. Konys, "Knowledge systematization for ontology learning methods," *Proc. Comput. Sci.*, vol. 126, pp. 2194–2207, Jan. 2018.

[12] S. Albukhitan, T. Helmy, and A. Alnazer, "Arabic ontology learning using deep learning," in *Proc. Int. Conf. Web Intell.*, 2017, pp. 1138–1142.

[13] A. Zouaq, D. Gasevic, and M. Hatala, "Towards open ontology learning and filtering," *Inf. Syst.*, vol. 36, no. 7, pp. 1064–1081, Nov. 2011.

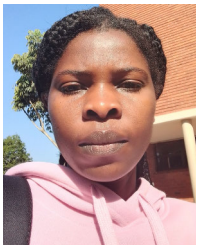
[14] M. Arguello Casteleiro, G. Demetriou, W. Read, M. J. Fernandez Prieto, N. Maroto, D. Maseda Fernandez, G. Nenadic, J. Klein, J. Keane, and R. Stevens, "Deep learning meets ontologies: Experiments to anchor the cardiovascular disease ontology in the biomedical literature," *J. Biomed. Semantics*, vol. 9, no. 1, pp. 1–24, Dec. 2018.

[15] R. Abada, A. M. Abubakar, and M. T. Bilal, "An overview on deep leaning application of big data," *Mesopotamian J. Big Data*, vol. 2022, pp. 31–35, Jul. 2022.
[16] M. M. Taye, "Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions," *Computers*, vol. 12, no. 5, p. 91, Apr. 2023.
[17] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019.
[18] R. Qamar and B. A. Zardari, "Artificial neural networks: An overview," *Mesopotamian J. Comput. Sci.*, vol. 2023, pp. 130–139, Aug. 2023.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.
[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[23] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
[24] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.
[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[26] OpenAI et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.
[27] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, and J. Schulman, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 27730–27744.
[28] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.
[29] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 8821–8831.
[30] S. J. Wei, D. F. Al Riza, and H. Nugroho, "Comparative study on the performance of deep learning implementation in the edge computing: Case study on the plant leaf disease identification," *J. Agricult. Food Res.*, vol. 10, Dec. 2022, Art. no. 100389.
[31] S. Chakraborty, F. M. J. M. Shamrat, M. Billah, A. Jubair, M. Alauddin, and R. Ranjan, "Implementation of deep learning methods to identify rotten fruits," in *Proc. 5th Int. Conf. Trends Electron. Informat. (ICOEI)*, Jun. 2021, pp. 1207–1212.
[32] F. Weidt and R. Silva, "Systematic literature review in computer science—A practical guide," *Relatórios Técnicos Do DCC/UFJF*, vol. 1, no. 8, pp. 1–7, 2016.
[33] F. Jauro, H. Chiroma, A. Y. Gital, M. Almutairi, S. M. Abdulhamid, and J. H. Abawajy, "Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106582.
[34] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *Ann. Internal Med.*, vol. 151, no. 4, pp. 264–269, 2009.
[35] A. Ayadi, A. Samet, F. De Bertrand de Beuvron, and C. Zanni-Merk, "Ontology population with deep learning-based NLP: A case study on the biomolecular network ontology," *Proc. Comput. Sci.*, vol. 159, pp. 572–581, Jan. 2019.
[36] M. Leimeister and B. J. Wilson, "Skip-gram word embeddings in hyperbolic space," 2018, *arXiv:1809.01498*.
[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[38] T. Adewumi, F. Liwicki, and M. Liwicki, "Word2 Vec: Optimal hyperparameters and their impact on natural language processing downstream tasks," *Open Comput. Sci.*, vol. 12, no. 1, pp. 134–141, Mar. 2022.
[39] M. Arguello Casteleiro, D. Maseda Fernandez, G. Demetriou, W. Read, M. J. Fernandez Prieto, J. Des Diz, G. Nenadic, J. Keane, and R. Stevens, "A case study on sepsis using PubMed and deep learning for ontology learning," in *Informatics for Health: Connected Citizen-Led Wellness and Population Health*. The Netherlands: IOS Press, 2017, pp. 516–520.
[40] B. Hu, B. Tang, Q. Chen, and L. Kang, "A novel word embedding learning model using the dissociation between nouns and verbs," *Neurocomputing*, vol. 171, pp. 1108–1117, Jan. 2016.
[41] H. Faouzi, M. El-Badaoui, M. Boutalline, A. Tannouche, and H. Ouanan, "Towards amazigh word embedding: Corpus creation and Word2 Vec models evaluations," *Revue d'Intell. Artificielle*, vol. 37, no. 3, pp. 753–759, Jun. 2023.
[42] S. Poornima and T. Subramanian, "Effective feature extraction via N-skip Gram instruction embedding model using deep neural network for designing anti-malware application," in *Proc. 9th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, vol. 1, Mar. 2023, pp. 2118–2123.
[43] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
[44] M. E. Peters, S. Ruder, and N. A. Smith, "To tune or not to tune? Adapting pretrained representations to diverse tasks," 2019, *arXiv:1903.05987*.
[45] M. E. Peters, M. Neumann, L. Zettlemoyer, and W. Yih, "Dissecting contextual word embeddings: Architecture and representation," 2018, *arXiv:1808.08949*.
[46] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," Tech. Rep., 2018. Accessed: Dec. 10, 2023. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
[47] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 19–27.
[48] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: A review of BERT-based approaches," *Artif. Intell. Rev.*, vol. 54, no. 8, pp. 5789–5829, Dec. 2021.
[49] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.
[50] E. T. R. Schneider, J. V. A. de Souza, Y. B. Gumiel, C. Moro, and E. C. Paraiso, "A GPT-2 language model for biomedical texts in Portuguese," in *Proc. IEEE 34th Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Jun. 2021, pp. 474–479.
[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[52] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proc. ACM Conf. Fairness, Accountability, Transparency*, 2021, pp. 610–623.
[53] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.
[54] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. Tat Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. Tulio Ribeiro, and Y. Zhang, "Sparks of artificial general intelligence: Early experiments with GPT-4," 2023, *arXiv:2303.12712*.
[55] I. Yelmen, A. Gunes, and M. Zontul, "Multi-class document classification using lexical ontology-based deep learning," *Appl. Sci.*, vol. 13, no. 10, p. 6139, May 2023.
[56] J. Shang, T. Ma, C. Xiao, and J. Sun, "Pre-training of graph augmented transformers for medication recommendation," 2019, *arXiv:1906.00346*.
[57] B. Srinivasan and B. Ribeiro, "On the equivalence between positional node embeddings and structural graph representations," 2019, *arXiv:1910.00452*.
[58] C. Jeong, S. Jang, E. Park, and S. Choi, "A context-aware citation recommendation model with BERT and graph convolutional networks," *Scientometrics*, vol. 124, no. 3, pp. 1907–1922, Sep. 2020.
[59] X. Wang, F. Huang, W. Wan, and C. Zhang, "Academic activities transaction extraction based on deep belief network," *Adv. Multimedia*, vol. 2017, no. 1, 2017, Art. no. 5067069.
[60] B. Zhong, J. Liu, Y. Du, Y. Liaozheng, and J. Pu, "Extracting attributes of named entity from unstructured text with deep belief network," *Int. J. Database Theory Appl.*, vol. 9, no. 5, pp. 187–196, May 2016.

[61] G. Zhong, L.-N. Wang, X. Ling, and J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *J. Finance Data Sci.*, vol. 2, no. 4, pp. 265–278, Dec. 2016.
[62] M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue, and R. Guan, "Text classification based on deep belief network and softmax regression," *Neural Comput. Appl.*, vol. 29, no. 1, pp. 61–70, Jan. 2018.
[63] W. Guoyu, *Research on Chinese Named Entity Recognition Based on Deep Learning*. Beijing, China: Beijing Univ. Technology, 2015, pp. 11–17.
[64] J. Yao, T. Sheng, J. Zhen, and X. Bao, "Fault prognosis based on restricted Boltzmann machine and data label for switching power amplifiers," in *Proc. 12th Int. Conf. Rel., Maintainability, Saf. (ICRMS)*, Oct. 2018, pp. 287–291.
[65] X. Dai, J. Cheng, Y. Gao, S. Guo, X. Yang, X. Xu, and Y. Cen, "Deep belief network for feature extraction of urban artificial targets," *Math. Problems Eng.*, vol. 2020, pp. 1–13, May 2020.
[66] D. Chen, J. Lv, and Z. Yi, "Graph regularized restricted Boltzmann machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2651–2659, Jun. 2018.
[67] Z. Li, H. Huang, Z. Zhang, and G. Shi, "Manifold-based multi-deep belief network for feature extraction of hyperspectral image," *Remote Sens.*, vol. 14, no. 6, p. 1484, Mar. 2022.
[68] Y. Feng, H. Zhang, W. Hao, and G. Chen, "(Named entity recognition based on deep belief net)," *Comput. Sci.*, vol. 43, no. 4, pp. 224–230, 2016.
[69] J. Yu and X. Yan, "Active features extracted by deep belief network for process monitoring," *ISA Trans.*, vol. 84, pp. 247–261, Jan. 2019.
[70] Y. Chen, "Research on Chinese information extraction based on deep belief nets," Ph.D. thesis, School Comput. Sci. Technol., Harbin Inst. Technol., Harbin, China, 2014.
[71] J. Leng and P. Jiang, "A deep learning approach for relationship extraction from interaction context in social manufacturing paradigm," *Knowl.-Based Syst.*, vol. 100, pp. 188–199, May 2016.
[72] P. Manda, S. SayedAhmed, and S. D. Mohanty, "Automated ontology-based annotation of scientific literature using deep learning," in *Proc. Int. Workshop Semantic Big Data*, 2020, pp. 1–6.
[73] P. Manda, L. Beasley, and S. D. Mohanty, "Taking a dive: Experiments in deep learning for automatic ontology-based annotation of scientific literature," *bioRxiv*, 2018, Art. no. 365874.
[74] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, "Detecting formal thought disorder by deep contextualized word representations," *Psychiatry Res.*, vol. 304, Oct. 2021, Art. no. 114135.
[75] A. Shaker, A. Aldarf, and I. Bessmertny, "Using LSTM and GRU with a new dataset for named entity recognition in the Arabic language," 2023, *arXiv:2304.03399*.
[76] N. Zafar, I. U. Haq, J.-U.-R. Chughtai, and O. Shafiq, "Applying hybrid LSTM-GRU model based on heterogeneous data sources for traffic speed prediction in urban areas," *Sensors*, vol. 22, no. 9, p. 3348, Apr. 2022.
[77] C. Yan, X. Fu, W. Wu, S. Lu, and J. Wu, "Neural network based relation extraction of enterprises in credit risk management," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2019, pp. 1–6.
[78] X. Dai, S. Karimi, B. Hachey, and C. Paris, "Using similarity measures to select pretraining data for NER," 2019, *arXiv:1904.00585*.
[79] R. Fan, L. Wang, J. Yan, W. Song, Y. Zhu, and X. Chen, "Deep learning-based named entity recognition and knowledge graph construction for geological hazards," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 1, p. 15, Dec. 2019.
[80] C. Lyu, B. Chen, Y. Ren, and D. Ji, "Long short-term memory RNN for biomedical named entity recognition," *BMC Bioinf.*, vol. 18, no. 1, pp. 1–11, Dec. 2017.
[81] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*.
[82] H. Wang, L. Zhang, H. Luo, J. He, and R. W. M. Cheung, "AI-powered landslide susceptibility assessment in Hong Kong," *Eng. Geol.*, vol. 288, Jul. 2021, Art. no. 106103.
[83] A. A. Aljarrah and A. H. Ali, "Human activity recognition using PCA and BiLSTM recurrent neural networks," in *Proc. 2nd Int. Conf. Eng. Technol. Appl. (IICETA)*, Aug. 2019, pp. 156–160.
[84] F. Pouromran, Y. Lin, and S. Kamarthi, "Personalized deep bi-LSTM RNN based model for pain intensity classification using EDA signal," *Sensors*, vol. 22, no. 21, p. 8087, Oct. 2022.
[85] T. A. Manoharan and M. Radhakrishnan, "Region-wise brain response classification of ASD children using EEG and BiLSTM RNN," *Clin. EEG Neurosci.*, vol. 54, no. 5, pp. 461–471, Sep. 2023.
[86] J. Oksanen, O. Cocarascu, and F. Toni, "Automatic product ontology extraction from textual reviews," 2021, *arXiv:2105.10966*.
[87] A. Harnoune, M. Rhanoui, M. Mikram, S. Yousfi, Z. Elkaimbillah, and B. El Asri, "BERT based clinical knowledge extraction for biomedical knowledge graph construction and analysis," *Comput. Methods Programs Biomed. Update*, vol. 1, Jan. 2021, Art. no. 100042.
[88] X. Yang, J. Bian, W. R. Hogan, and Y. Wu, "Clinical concept extraction using transformers," *J. Amer. Med. Inform. Assoc.*, vol. 27, no. 12, pp. 1935–1942, Dec. 2020.
[89] Y. Si, J. Wang, H. Xu, and K. Roberts, "Enhancing clinical concept extraction with contextual embeddings," *J. Amer. Med. Inform. Assoc.*, vol. 26, no. 11, pp. 1297–1304, Nov. 2019.
[90] Y. Fang, D. Zhang, and G. Wu, "Toward establishing a knowledge graph for drought disaster based on ontology design and named entity recognition," *J. Hydroinform.*, vol. 25, no. 4, pp. 1457–1470, Jul. 2023.
[91] T. Jinjun, T. Haonan, L. You, and F. Qiang, "Autonomous transportation system participant identification method based on BERT-bi-LSTM-CRF model," *Traffic Inf. Saf.*, vol. 40, pp. 80–90, Oct. 2022.
[92] H. Xu, G. Fan, G. Kuang, and C. Wang, "Exploring the potential of BERT-BiLSTM-CRF and the attention mechanism in building a tourism knowledge graph," *Electronics*, vol. 12, no. 4, p. 1010, Feb. 2023.
[93] J. Liu, C. Sun, and Y. Yuan, "The BERT-BiLSTM-CRF question event information extraction method," in *Proc. IEEE 3rd Int. Conf. Electron. Inf. Commun. Technol. (ICEICT)*, Nov. 2020, pp. 729–733.
[94] Y. Ren, J. Han, Y. Lin, X. Mei, and L. Zhang, "An ontology-based and deep learning-driven method for extracting legal facts from Chinese legal texts," *Electronics*, vol. 11, no. 12, p. 1821, Jun. 2022.
[95] M. Aydar, O. Bozal, and F. Ozbay, "Neural relation extraction: A survey," 2020, *arXiv:2007.04247*.
[96] B. Ganesan, R. Dasgupta, A. Parekh, H. Patel, and B. Reinwald, "A neural architecture for person ontology population," 2020, *arXiv:2001.08013*.
[97] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.
[98] A. Lamurias, D. Sousa, L. A. Clarke, and F. M. Couto, "BO-LSTM: Classifying relations via long short-term memory networks along biomedical ontologies," *BMC Bioinf.*, vol. 20, no. 1, pp. 1–12, Dec. 2019.
[99] D. Sousa, "Deep learning system for biomedical relation extraction combining external sources of knowledge," in *Proc. Eur. Conf. Inf. Retr.* Cham, Switzerland: Springer, 2021, pp. 688–693.
[100] S. I. R. Conceição and F. M. Couto, "Text mining for building biomedical networks using cancer as a case study," *Biomolecules*, vol. 11, no. 10, p. 1430, Sep. 2021.
[101] H. Wu, Y. Xing, W. Ge, X. Liu, J. Zou, C. Zhou, and J. Liao, "Drug-drug interaction extraction via hybrid neural networks on biomedical literature," *J. Biomed. Informat.*, vol. 106, Jun. 2020, Art. no. 103432.
[102] D. Sousa, A. Lamurias, and F. M. Couto, "Using neural networks for relation extraction from biomedical literature," in *Artificial Neural Networks*. New York, NY, USA: Springer, 2020, pp. 289–305.
[103] D. Sousa and F. M. Couto, "BiOnt: Deep learning using multiple biomedical ontologies for relation extraction," in *Proc. Eur. Conf. Inf. Retr.* Cham, Switzerland: Springer, 2020, pp. 367–374.
[104] B. Xu, X. Shi, Z. Zhao, and W. Zheng, "Leveraging biomedical resources in bi-LSTM for drug-drug interaction extraction," *IEEE Access*, vol. 6, pp. 33432–33439, 2018.
[105] F. Ali, S. El-Sappagh, and D. Kwak, "Fuzzy ontology and LSTM-based text mining: A transportation network monitoring system for assisting travel," *Sensors*, vol. 19, no. 2, p. 234, Jan. 2019.
[106] F. Dhombres and J. Charlet, "Design and use of semantic resources: Findings from the section on knowledge representation and management of the 2020 international medical informatics association yearbook," *Yearbook Med. Informat.*, vol. 29, no. 1, pp. 163–168, Aug. 2020.
[107] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, Sep. 2015, pp. 1753–1762.
[108] Z. Kuang, J. Yu, Z. Li, B. Zhang, and J. Fan, "Integrating multi-level deep learning and concept ontology for large-scale visual recognition," *Pattern Recognit.*, vol. 78, pp. 198–214, Jun. 2018.
[109] S. Zheng, J. Xu, P. Zhou, H. Bao, Z. Qi, and B. Xu, "A neural network framework for relation extraction: Learning entity semantic and relation pattern," *Knowl.-Based Syst.*, vol. 114, pp. 12–23, Dec. 2016.
[110] J. Wang, J. Liu, and L. Kong, "Ontology construction based on deep learning," in *Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 17*. Singapore: Springer, 2018, pp. 505–510.

[111] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.
[112] W. H. Bangyal, N. U. Rehman, A. Nawaz, K. Nisar, A. A. A. Ibrahim, R. Shakir, and D. B. Rawat, "Constructing domain ontology for Alzheimer disease using deep learning based approach," *Electronics*, vol. 11, no. 12, p. 1890, Jun. 2022.
[113] S. R. Guruvayur and R. Suchithra, "Automatic relationship construction in domain ontology engineering using semantic and thematic graph generation process and convolution neural network," *Int. J. Recent Technol. Eng. (IJRTE)*, vol. 8, no. 3, pp. 4602–4610, Sep. 2019.
[114] D. Tian, M. Li, J. Shi, Y. Shen, and S. Han, "On-site text classification and knowledge mining for large-scale projects construction by integrated intelligent approach," *Adv. Eng. Informat.*, vol. 49, Aug. 2021, Art. no. 101355.
[115] P. S. Nair, K. R. Rao, and M. S. Nair, "A machine learning approach for fast mode decision in HEVC intra prediction based on statistical features," *J. Intell. Fuzzy Syst.*, vol. 36, no. 3, pp. 2095–2106, Mar. 2019.
[116] H. Fu, Z. Niu, C. Zhang, J. Ma, and J. Chen, "Visual cortex inspired CNN model for feature construction in text analysis," *Frontiers Comput. Neurosci.*, vol. 10, p. 64, Jul. 2016.
[117] Q. Guo, F. Wang, J. Lei, D. Tu, and G. Li, "Convolutional feature learning and hybrid CNN-HMM for scene number recognition," *Neurocomputing*, vol. 184, pp. 78–90, Apr. 2016.
[118] Q. Hu, Q. Li, Y. Lu, Y. Yang, and J. Cheng, "Multi-level word features based on CNN for fake news detection in cultural communication," *Pers. Ubiquitous Comput.*, vol. 24, no. 2, pp. 259–272, Apr. 2020.
[119] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," 2016, *arXiv:1606.01781*.
[120] Z. Geng, J. Li, Y. Han, and Y. Zhang, "Novel target attention convolutional neural network for relation classification," *Inf. Sci.*, vol. 597, pp. 24–37, Jun. 2022.
[121] M. R. Hossain and M. M. Hoque, "Automatic Bengali document categorization based on deep convolution nets," in *Emerging Research in Computing, Information, Communication and Applications: ERCICA*, vol. 1. Singapore: Springer, 2019, pp. 513–525.
[122] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. COLING, 25th Int. Conf. Comput. Linguistics: Tech. Papers*, 2014, pp. 2335–2344.
[123] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," 2016, *arXiv:1610.08815*.
[124] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. COLING, 25th Int. Conf. Comput. Linguistics: Tech. Papers*, 2014, pp. 69–78.
[125] T. Nayak, "Deep neural networks for relation extraction," Ph.D. thesis, Dept. Comput. Sci., School Comput. Nat. Univ. Singapore, Singapore, 2021.
[126] Z. Hao, W. Mayer, J. Xia, G. Li, L. Qin, and Z. Feng, "Ontology alignment with semantic and structural embeddings," *J. Web Semantics*, vol. 78, Oct. 2023, Art. no. 100798.
[127] N. Zhang, S. Deng, Z. Sun, G. Wang, X. Chen, W. Zhang, and H. Chen, "Long-tail relation extraction via knowledge graph embeddings and graph convolution networks," 2019, *arXiv:1903.01306*.
[128] X. Jiang, Q. Wang, P. Li, and B. Wang, "Relation extraction with multi-instance multi-label convolutional neural networks," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics: Tech. Papers*, 2016, pp. 1471–1480.
[129] D. Zhang and D. Wang, "Relation classification via recurrent neural network," 2015, *arXiv:1508.01006*.
[130] H. Tang, Y. Mi, F. Xue, and Y. Cao, "An integration model based on graph convolutional network for text classification," *IEEE Access*, vol. 8, pp. 148865–148876, 2020.
[131] Y. Meng, J. Shen, C. Zhang, and J. Han, "Weakly-supervised neural text classification," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 983–992.
[132] Y. Li, X. Wang, and P. Xu, "Chinese text classification model based on deep learning," *Future Internet*, vol. 10, no. 11, p. 113, Nov. 2018.
[133] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 3, pp. 912–921, May 2021.
[134] J. Zhang, P. Lertvittayakumjorn, and Y. Guo, "Integrating semantic knowledge to tackle zero-shot text classification," 2019, *arXiv:1903.12626*.
[135] X. Li, X. Wu, Z. Luo, Z. Du, Z. Wang, and C. Gao, "Integration of global and local information for text classification," *Neural Comput. Appl.*, vol. 35, no. 3, pp. 2471–2486, Jan. 2023.
[136] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, "User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106435.
[137] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional long short-term memory networks for relation classification," in *Proc. 29th Pacific Asia Conf. Lang., Inf. Comput.*, 2015, pp. 73–78.
[138] H. Wang, C. Focke, R. Sylvester, N. Mishra, and W. Wang, "Fine-tune BERT for DocRED with two-step process," 2019, *arXiv:1909.11898*.
[139] L. Gomes, M. Côrtes, and R. Torres, "BERT-based feature extraction for long-lived bug prediction in floss: A comparative study," *Inf. Softw. Technol.*, vol. 160, Aug. 2022, Art. no. 107217.
[140] X. Han and L. Wang, "A novel document-level relation extraction method based on BERT and entity information," *IEEE Access*, vol. 8, pp. 96912–96919, 2020.
[141] Y. Papanikolaou, I. Roberts, and A. Pierleoni, "Deep bidirectional transformers for relation extraction without supervision," 2019, *arXiv:1911.00313*.
[142] H. Liu, Y. Perl, and J. Geller, "Concept placement using BERT trained by transforming and summarizing biomedical ontology structure," *J. Biomed. Informat.*, vol. 112, Dec. 2020, Art. no. 103607.
[143] H. Liu, Y. Perl, and J. Geller, "Transfer learning from BERT to support insertion of new concepts into SNOMED CT," in *Proc. AMIA Annu. Symp.*, 2019, p. 1129.
[144] M. Barbouch, S. Verberne, and T. Verhoef, "WN-BERT: Integrating wordnet and BERT for lexical semantics in natural language understanding," *Comput. Linguistics Netherlands J.*, vol. 11, pp. 105–124, Dec. 2021.
[145] L. Huang, C. Sun, X. Qiu, and X. Huang, "GlossBERT: BERT for word sense disambiguation with gloss knowledge," 2019, *arXiv:1908.07245*.
[146] Z. Lu, P. Du, and J.-Y. Nie, "VGCN-BERT: Augmenting BERT with graph embedding for text classification," in *Proc. Adv. Inf. Retr., 42nd Eur. Conf. IR Res. (ECIR)*, Lisbon, Portugal. Cham, Switzerland: Springer, 2020, pp. 369–382.
[147] B. Xue, C. Zhu, X. Wang, and W. Zhu, "An Integration Model for Text Classification using Graph Convolutional Network and BERT," in *Proc. J. Phys., Conf.*, vol. 2137, 2021, Art. no. 012052.
[148] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–8.
[149] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[150] B. Xue, C. Zhu, X. Wang, and W. Zhu, "The study on the text classification based on graph convolutional network and BiLSTM," in *Proc. 8th Int. Conf. Comput. Artif. Intell.*, 2022, pp. 323–331.
[151] S. Neutel and M. H. de Boer, "Towards automatic ontology alignment using BERT," in *Proc. AAAI Spring Symp., Combining Mach. Learn. Knowl. Eng.*, 2021, pp. 1–12.
[152] M. S. M. Rudwan and J. V. Fonou-Dombeu, "Hybridizing fuzzy string matching and machine learning for improved ontology alignment," *Future Internet*, vol. 15, no. 7, p. 229, Jun. 2023.
[153] Y. He, J. Chen, D. Antonyrajah, and I. Horrocks, "BERTMap: A BERT-based ontology alignment system," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 5684–5691.
[154] Y. He, J. Chen, H. Dong, I. Horrocks, C. Allocca, T. Kim, and B. Sapkota, "DeepOnto: A Python package for ontology engineering with deep learning," 2023, *arXiv:2307.03067*.
[155] B. Jiménez Gutiérrez, N. McNeal, C. Washington, Y. Chen, L. Li, H. Sun, and Y. Su, "Thinking about GPT-3 in-context learning for biomedical IE? Think again," 2022, *arXiv:2203.08410*.
[156] P. Bellan, M. Dragoni, and C. Ghidini, "Extracting business process entities and relations from text using pre-trained language models and in-context learning," in *Proc. Int. Conf. Enterprise Des., Oper., Comput.* Cham, Switzerland: Springer, 2022, pp. 182–199.
[157] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, "Large language models are few-shot clinical information extractors," 2022, *arXiv:2205.12689*.
[158] T. Groza, H. Caufield, D. Gration, G. Baynam, M. A. Haendel, P. N. Robinson, C. J. Mungall, and J. T. Reese, "An evaluation of GPT models for phenotype concept recognition," *BMC Med. Informat. Decis. Making*, vol. 24, no. 1, p. 30, Jan. 2024.
[159] Y. Qiu and Y. Jin, "ChatGPT and finetuned BERT: A comparative study for developing intelligent design support systems," *Intell. Syst. Appl.*, vol. 21, Mar. 2024, Art. no. 200308.
[160] G. Petrucci, C. Ghidini, and M. Rospocher, "Using recurrent neural network for learning expressive ontologies," 2016, *arXiv:1607.04110*.

[161] B. Peng, Z. Lu, H. Li, and K.-F. Wong, "Towards neural network-based reasoning," 2015, *arXiv:1508.05508*.
[162] C.-H. Cai, D. Ke, Y. Xu, and K. Su, "Symbolic manipulation based on deep neural networks and its application to axiom discovery," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2136–2143.
[163] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020.
[164] F. Alam, H. B. Giglou, and K. M. Malik, "Automated clinical knowledge graph generation framework for evidence based medicine," *Expert Syst. Appl.*, vol. 233, Dec. 2023, Art. no. 120964.
[165] N. Chandolikar, S. Shilaskar, D. Peddawad, and S. Bhosale, "Semi-automated ontology building using deep learning to provide domain-specific knowledge search in the Marathi language," in *Proc. Int. Conf. Appl. Mach. Learn. (ICAML)*, May 2019, pp. 108–113.
[166] N. Abdelmageed, F. Löffler, and B. König-Ries, "BiodivBERT: A pre-trained language model for the biodiversity domain," Tech. Rep., 2023.
[167] Z. Zhao, Z. Yang, L. Luo, H. Lin, and J. Wang, "Drug drug interaction extraction from biomedical literature using syntax convolutional neural network," *Bioinformatics*, vol. 32, no. 22, pp. 3444–3453, Nov. 2016.
[168] M. Jain, K. Singh, and R. Mutharaju, "ReOnto: A neuro-symbolic approach for biomedical relation extraction," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2023, pp. 230–247.
[169] M. A. Casteleiro, M. J. F. Prieto, G. Demetriou, N. Maroto, W. J. Read, D. Maseda-Fernandez, J. J. Des Diz, G. Nenadic, J. A. Keane, and R. Stevens, "Ontology learning with deep learning: A case study on patient safety using PubMed," in *Proc. SWAT4LS*, 2016, pp. 1–10.
[170] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
[171] G. Petrucci, C. Ghidini, and M. Rospocher, "Ontology learning in the deep," in *Proc. Eur. Knowl. Acquisition Workshop*. MIT Press, 2016, pp. 480–495.
[172] Y. Ma and A. Syamsiyah, "A hybrid approach to learn description logic based biomedical ontology from texts," in *Proc. Int. Conf. Posters Demonstrations Track*, vol. 1272, 2014, pp. 421–424.
[173] R. Kiani, A. Keshavarzi, and M. Bohlouli, "Detection of thin boundaries between different types of anomalies in outlier detection using enhanced neural networks," *Appl. Artif. Intell.*, vol. 34, no. 5, pp. 345–377, Apr. 2020.
[174] Y. Guo, S. Li, R. Deng, L. Qi, H. Zhang, and Y. Li, "Multi-dimensional outliers detection method based on RBF neural network model," *Revista Ibérica de Sistemas e Tecnologias de Informação*, no. E9, p. 21, 2016.
[175] H. Chiroma, A. Y. Gital, N. Rana, S. M. Abdulhamid, A. N. Muhammad, A. Y. Umar, and A. I. Abubakar, "Nature inspired meta-heuristic algorithms for deep learning: Recent progress and novel perspective," in *Proc. Adv. Comput. Vis., Comput. Vis. Conf. (CVC)*, vol. 1. Cham, Switzerland: Springer, 2020, pp. 59–70.
[176] A. C. Fang, W.-Y. Li, and J. Cao, "A corpus-based approach to fingerprinting stylistic features of classical chinese poetry: A Case Study of Liu Yong and Su Shi," *J. Chin. Linguistics Monograph Ser.*, vol. 25, pp. 63–81, 2015.



TSITSI ZENGEYA received the B.Sc. degree (Hons.) in computer science from Bindura University of Science Education (BUSE), Zimbabwe, and the M.Sc. degree in computer science from the National University of Science and Technology (NUST), Zimbabwe. She is currently pursuing the Ph.D. degree with the University of KwaZulu-Natal (UKZN), South Africa. Her research interests include ontologies, machine learning, and deep learning.

JEAN VINCENT FONOU-DOMBEU received the B.Sc. (Hons.) and B.Sc. degrees in computer science from the University of Yaoundé I, Cameroon, the M.Sc. degree in computer science from the University of KwaZulu-Natal, South Africa, and the Ph.D. degree in computer science from North-West University, South Africa. He is currently a Senior Lecturer with the Department of Computer Science, University of KwaZulu-Natal (UKZN). His research interests include ontology engineering, semantic web, natural language processing, machine learning, and their applications; specifically, in ontology building, learning, modularization, ranking, summarization and visualization, artificial intelligence, machine learning and data mining methods for the semantic web, knowledge representation and reasoning on the web, and knowledge graphs and deep semantics.