Neurocomputing 616 (2025) 128921

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# A survey of text classification based on pre-trained language model

Yujia Wu [a,b], Jun Wan [c,*]

[a] School of Information Science and Technology, Sanda University, Shanghai, 201209, China
[b] School of Computer Science, Wuhan University, Wuhuan, 430072, China
[c] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, 430073, China

ARTICLE INFO

*Keywords:*
Machine learning
Deep learning
Neural networks
Natural language processing
Text classification
Transformer
Pre-trained language models

ABSTRACT

The utilization of text classification is widespread within the domain of Natural Language Processing (NLP). In recent years, pre-trained language models (PLMs) based on the Transformer architecture have made significant strides across various artificial intelligence tasks. Currently, text classification employing PLMs has emerged as a prominent research focus within NLP. While several review papers examine text classification and Transformer models, there is a notable lack of comprehensive surveys specifically addressing text classification grounded in PLMs. To address this gap, the present survey provides an extensive overview of text classification techniques that leverage PLMs. The primary components of this review include: (1) an introduction, (2) a systematic examination of PLMs, (3) deep learning-based text classification methodologies, (4) text classification approaches utilizing pre-trained models, (5) commonly used datasets and evaluation metrics in text classification, (6) prevalent challenges and emerging trends in the field, and (7) a conclusion.

## 1. Introduction

With the rapid advancement of artificial intelligence technology [1–4], text classification has emerged as a pivotal technique in a wide range of applications [5]. Text classification is a technique that has been widely applied in information retrieval, data mining, and Natural Language Processing (NLP). To execute the task of text classification, users must first define the classes and extract relevant text features to construct an effective text classifier [6]. This process entails analyzing and categorizing text content, subsequently assigning one or more class labels. For instance, sentiment analysis is employed to evaluate and rate the sentiment expressed in product or movie reviews, while topic analysis is utilized to classify the themes of texts, such as those related to sports or education [7,8]. Text classification can be broadly categorized into document classification and sentence classification [9], depending on the granularity of the objects being processed [10]. Fundamentally, the primary objective of text classification is to employ computational tools to achieve human-level judgment capabilities, thereby enabling the accurate identification of a given text within a specific class [11]. Owing to its extensive applications in spam classification, question answering, information retrieval, sentiment analysis, and recommendation systems, text classification has rapidly emerged as a prominent topic within the field of NLP [12–15].

The processing flow of text classification encompasses three primary stages: text preprocessing, text feature extraction, and the construction of classification models [16–18]. A significant challenge within this framework is the extraction of text features while maintaining the integrity of semantic information [19–22]. Recently, text classification methods leveraging pre-trained language models (PLMs) have emerged as the predominant approach [23–25]. The primary advantage of this methodology lies in its ability to utilize external knowledge from PLMs, which enhances semantic understanding and improves the accuracy of text classification [26–28]. Among the most notable PLMs is BERT [29], a general-purpose language model trained on extensive text corpora, including Wikipedia. BERT's training process involves optimization through two key tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP).

This survey focuses on text classification techniques that utilize PLMs [26]. Fig. 1 illustrates a typical text classification model based on these pre-trained models. Specifically, PLMs grounded in the Transformer architecture can accept one or more labeled sequences as input, enabling text classification models to employ classification tokens $[CLS]$ as class feature vectors without necessitating structural modifications [30,31]. Following a straightforward linear layer, these models can effectively accomplish text classification tasks, resulting in two distinct classes, as depicted in Fig. 1.

Furthermore, by extracting features from PLMs and subsequently constructing specialized classification models for downstream tasks, researchers can leverage the general knowledge embedded within PLMs and fine-tune these models to enhance their performance in specific

* Corresponding author.
*E-mail address:* junwan2014@whu.edu.cn (J. Wan).

https://doi.org/10.1016/j.neucom.2024.128921
Received 14 March 2024; Received in revised form 30 September 2024; Accepted 10 November 2024
Available online 15 November 2024

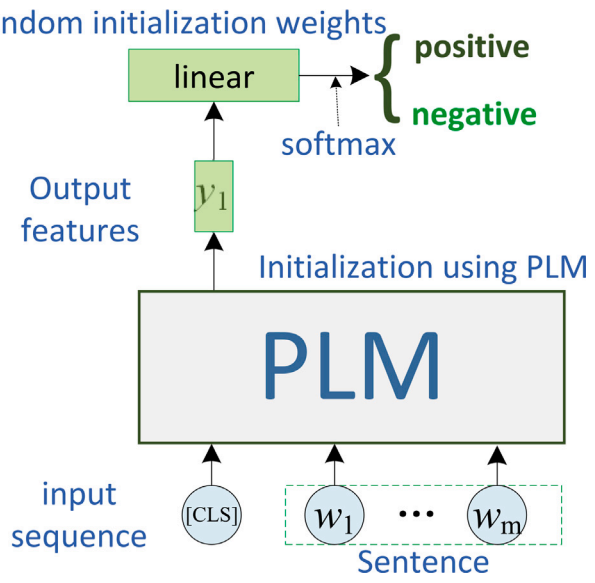


**Fig. 1.** A typical model for text classification based on PLMs.

applications [32]. In recent years, a variety of text classification techniques based on PLMs have emerged [33], facilitating their application across diverse fields [34]. This proliferation of methods has led to significant advancements in text classification research and practical implementations, underscoring the need for a timely review and summarization of representative techniques. Such a review will enable researchers to gain a comprehensive understanding of the current landscape of PLM utilization in text classification, as well as the achievements and challenges that remain.

Numerous reviews of text classification have been conducted from various perspectives. For instance, Bayer et al. [35] provide an extensive review of data augmentation methods in text classification, categorizing over 100 techniques into 12 distinct groups. Liu et al. [36] present a comprehensive overview of hierarchical multi-label text classification, while Wang et al. [17] focus on text classification methods based on graph neural networks. Additionally, Costa et al. [37] examine the application of embedding methods in text classification. Despite the existence of several review papers addressing the application of deep learning in text classification [12,17,19,35–37] and pre-trained language models [38–42], there remains a scarcity of comprehensive reviews that explore the combined application of these two domains.

The objective of this survey is to present a thorough summary and analysis of the advancements and trends in utilizing PLMs for text classification tasks. This survey builds upon existing literature in two specific dimensions: text classification and Transformer architectures. Unlike reviews that focus on conventional machine learning models, this survey emphasizes the application of PLMs based on the Transformer framework. It introduces classical deep learning text classification methods alongside those based on PLMs. This endeavor has the potential to significantly advance the field of text classification, given its prominence as one of the most extensively studied topics in NLP [5,15].

This survey aims to comprehensively examine the specific design aspects of PLMs that utilize the Transformer architecture. The discussion will encompass well-known PLMs [43], such as BERT [29] and GPT [44]. While recent surveys have explored various facets of the Transformer, including general Transformers [38], Visual Transformers [45], and Video Transformers [46], there are few comprehensive reviews that specifically address text classification based on PLMs. Although some surveys consider text classification [12,17,19,35–37], their scope, classification, and coverage are often limited, failing to encompass the common methods for text classification that rely on PLMs. To the best of our knowledge, no existing survey focuses exclusively on text classification based on PLMs. Consequently, this survey will concentrate on the intersection of PLMs and Transformers, exploring their applications and developmental trends within the domain of text classification.

This survey presents a thorough review of text classification techniques based on PLMs. The primary features of this survey include the following: **(1)** highlighting the advantages of PLMs in text feature extraction and their compatibility with diverse text classification methods, while also elucidating the intrinsic characteristics of PLMs from the perspective of text feature representation; **(2)** introducing classical deep learning-based text classification methods that possess the potential to significantly enhance classification performance; and **(3)** examining how PLMs grounded in the Transformer architecture are processed through self-attention mechanisms and their variants.

The survey comprises the following main sections:

(1) An introduction to the significance of text classification and the role of PLMs in this context.
(2) A systematic review of PLMs, presenting various types and categories.
(3) An overview of the application of conventional deep learning algorithms in text classification.
(4) A detailed exploration of how to effectively employ PLMs for text classification tasks.
(5) A discussion of common datasets and evaluation metrics utilized in text classification research.
(6) An examination of the current challenges faced in text classification, along with a discourse on future development trends and research directions.
(7) A conclusion summarizing the primary content and contributions of this survey, while outlining potential avenues for future research.

## 2. Pre-trained language model

Inspired by the success of ImageNet in the field of computer vision [47], the domain of NLP has increasingly leveraged PLMs for a variety of tasks. The emergence of dynamic word embedding models, such as ELMo [48], has paved the way for the development of PLMs. Furthermore, PLMs based on the Transformer framework, such as BERT [29], have initiated a new era of fine-tuning paradigms in NLP. Typically, the methodology for utilizing PLMs involves two primary steps: first, a general model is trained on a large dataset until it achieves satisfactory performance [49,50]; second, the PLM is fine-tuned and adapted for specific tasks using targeted datasets [51–53]. With the rapid advancement of PLMs, they have become the predominant approach for addressing downstream NLP tasks.

### 2.1. Transformer

The attention mechanism was first proposed by Mnih et al. [54] in 2014 and was initially applied in the realm of computer vision. The authors introduced the attention mechanism to recurrent neural networks (RNNs) for image classification, resulting in significant performance improvements. In 2015, Bahdanau et al. [55] further adapted the attention mechanism for NLP, employing it in machine translation, which similarly yielded noteworthy enhancements. The Transformer model, introduced by Vaswani et al. [56], completely discarded traditional network structures such as RNNs and convolutional neural networks (CNNs), relying exclusively on self-attention mechanisms [57–59]. This model can operate in parallel and effectively capture long-range dependencies, rendering it a powerful tool for various NLP tasks.

Fig. 2 shows the framework of Transformer. The Transformer model is constructed using an encoder–decoder architecture, which consists of

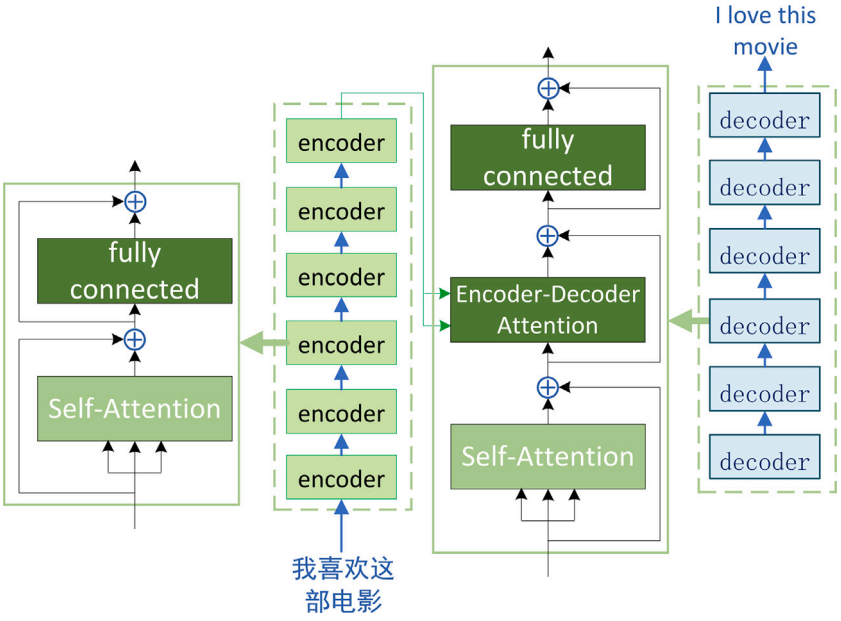


**Fig. 2.** The Transformer model consists of multiple encoders and multiple decoders.

two primary subnetworks: a self-attention module and a feedforward neural network. The self-attention module employs a multi-head attention mechanism, allowing the encoder to focus on distinct segments of the input sequence. Additionally, each subnetwork incorporates residual connections, which facilitate the retention of information across layers. The output of the encoder is a normalized feature vector that is subsequently processed by the decoder. The decoder comprises a self-attention module, a fully connected feedforward network, and an additional "encoder–decoder attention" module. This attention mechanism is interposed between the two subnetworks, enabling the decoder to attend to relevant portions of the encoded input sequence. Similar to the encoder, both modules in the decoder utilize residual connections and normalization operations to optimize information flow and mitigate the vanishing gradient problem. The mathematical representation of the residual connection and normalization operations in both the encoder and decoder can be expressed as follows:

$$y = layerNorm(Sublayer(x) + x) \tag{1}$$

where $layerNorm(\cdot)$ denotes feature normalization operations.

The Residual Network (ResNet) [60] was developed to address the problem of degradation, which arises from challenges during training in deep CNN models. For a stack of layers, ResNet learns the residual $F(x) = H(x) - x$, where $H(x)$ denotes the learned feature and $x$ denotes the input. When $F(x) = 0$, $H(x) = x$, indicating that residual learning is simpler than learning raw features directly. In such cases, the stacked layers perform solely identity mappings, and network performance will not deteriorate. However, in practice, the residual is non-zero, enabling the stacked layers to learn novel features based on the input features and achieve superior performance. By introducing residual connections, ResNet facilitates the training of extremely deep networks and has attained better performance on a diverse range of tasks. Fig. 3 illustrates the residual learning unit.

The Transformer model employs the self-attention mechanism to compute the output of the self-attention layer, as illustrated in Eq. (2):

$$Attention(Q, K, V) = softmax\left(\frac{Q \times K^T}{\sqrt{d_k}}\right)V \tag{2}$$

Here, $Q$, $K$, and $V$ represent the query, key, and value matrices, respectively. The dot product of $Q$ and $K$ is normalized by dividing it by the square root of the key dimension (which is also the query dimension), denoted as $d_k$. This normalization prevents the dot product result from becoming excessively scattered when $d_k$ is large.

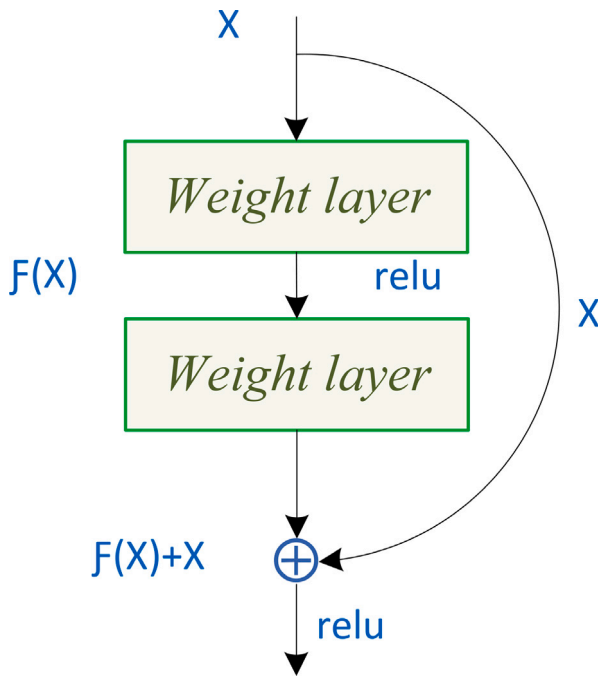


**Fig. 3.** Residual learning unit.

The multi-head mechanism in the self-attention layer offers multiple expressive pathways. Rather than a single set of $Q$, $K$, and $V$ matrices, there are multiple sets under multi-head. The linear layer does not feature a non-linear activation layer, and a single fully connected neural network processes other sequence inputs in the same manner. These network parameters are shared, reducing the number of parameters required and improving model efficiency.

However, the structure of the self-attention mechanism lacks the sequential information necessary to capture the order of words in a sentence. If the order of words is altered, the features captured by the

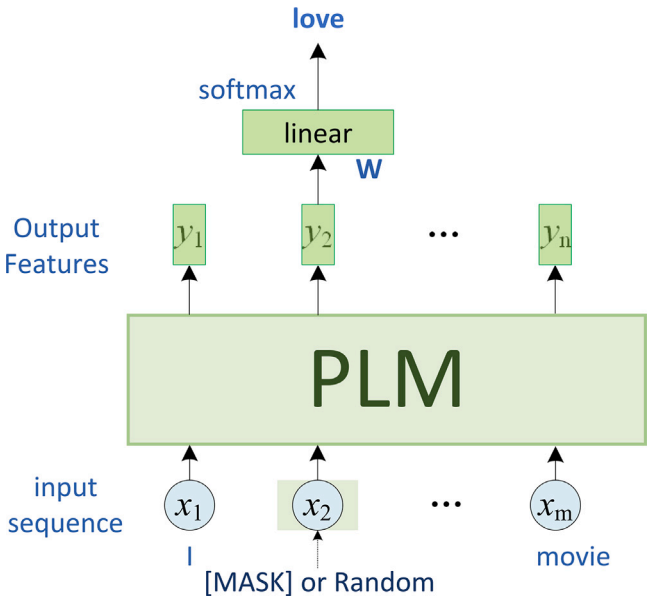


**Fig. 4.** The MLM task of BERT model.

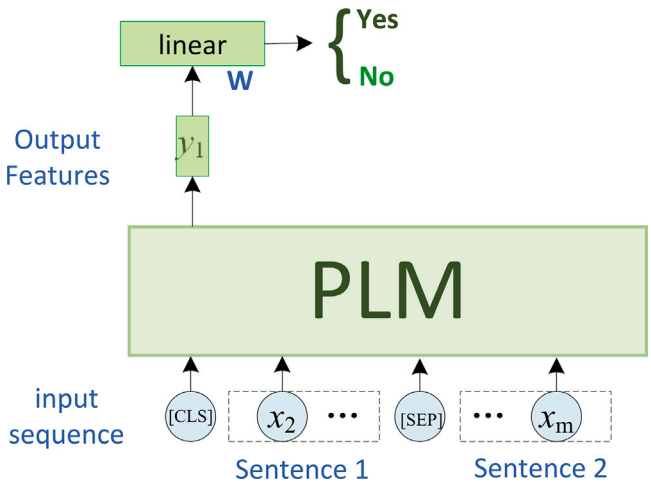


**Fig. 5.** The NSP task of BERT model.

Transformer remain unchanged, which is unsuitable for certain NLP tasks. To address this limitation, the Transformer integrates positional embeddings into its word embedding layer to reflect the positional relationships between words in a text sequence [56]. This approach allows the Transformer to effectively encode the order of words, thereby enhancing its applicability to a wider range of NLP tasks.

### 2.2. BERT

The BERT model [29] represents one of the most influential advancements in the domain of PLMs. Through a self-supervised training approach, BERT enables the efficient acquisition of substantial semantic knowledge, demonstrating remarkable performance across a variety of NLP tasks, including machine translation, text classification, and text similarity analysis. Currently, BERT remains one of the most widely adopted PLMs [61]. Its architecture is based on the Transformer encoder, constructing a bidirectional deep language model. Furthermore, BERT integrates Masked Language Modeling and Next Sentence Prediction for joint bidirectional training. During the pre-training phase, MLM plays a pivotal role, as illustrated in Fig. 4, allowing BERT to capture and comprehend bidirectional contextual semantic information, thereby enhancing the model's overall performance.

In the pre-training phase of the MLM, BERT employs a random masking strategy, wherein specific words are replaced with a designated token, $[MASK]$. The model utilizes contextual cues to predict these masked words, thereby capturing and understanding contextual information. This training approach, referred to as self-supervised learning, is a variant of unsupervised learning that employs supervised-like techniques by leveraging information within the dataset itself to create pseudo-labels.

Establishing semantic relationships between sentences is crucial for various downstream NLP tasks. To facilitate the acquisition of semantic connections between sentences, BERT also incorporates the NSP task during pre-training. Fig. 5 illustrates the training methodology of the NSP task, which involves determining whether a semantic connection exists between two sentences. The $[SEP]$ token serves as a delimiter; if a semantic connection exists between the two sentences, the output is 1; otherwise, the output is 0.

Despite BERT's significant achievements in NLP tasks, certain limitations persist in its application. The model exhibits a slower convergence rate during training, which may stem from inconsistencies between its pre-training and generation processes. Consequently, BERT's performance may be suboptimal in certain natural language generation tasks, and it may not be well-suited for document-level NLP tasks, limiting its applicability primarily to sentence-level and paragraph-level tasks. To address these limitations and enhance performance in PLMs, researchers have proposed models such as AlBERT [62], which reduces model size by sharing parameters across layers and replaces the NSP task with the Sentence Order Prediction task, thereby improving overall model performance.

The ERNIE model [63] represents a knowledge-enhanced semantic representation framework that comprises two primary modules: the lower layer and the upper layer. The lower module functions as a text encoder, capturing fundamental vocabulary and sentence information from the input. In contrast, the upper module serves as a knowledge encoder, integrating knowledge information extracted from the lower layer into the text information. ERNIE's training methodology uniformly models syntactic, lexical, and semantic information present in the training data, leveraging both MLM and NSP pre-training tasks to extract relevant information.

ELECTRA [64] addresses the inconsistency issue observed in BERT during the pre-training and fine-tuning stages by substituting MLM with token detection. This approach not only enhances the computational efficiency of the model but also improves its absolute performance. Similarly, MPNet [65] effectively utilizes the dependencies between predicted labels through permutation language modeling, while also incorporating auxiliary position information as input. This enables the model to perceive the entire sentence while minimizing the impact of positional differences.

For models based on the Transformer framework, the input sequence length is directly proportional to the model's complexity. As the sequence length increases, the model requires a substantial amount of memory, which can reduce computational efficiency. To address this challenge, Dai et al. [66] proposed the Funnel Transformer, which tapers in the sequence direction as the number of layers increases, thereby economizing space overhead. Its training methodology is congruent with that of BERT, allowing for efficient processing of longer sequences while maintaining performance.

### 2.3. Transformer-XL

The standard Transformer model processes the entire sequence in parallel, utilizing self-attention mechanisms to capture relationships between sequence elements and generating the output sequence in a

single pass. To facilitate this parallel processing, the model is restricted to fixed-length sequence inputs. Sequences that are insufficiently long require padding, while those that exceed the maximum length must be truncated. To address these challenges, Al-Rfou et al. [67] proposed a 64-layer Transformer decoder architecture for a character-level language model in 2018. This model employs a "segmented training plus sliding window inference" approach. While this method partially alleviates the issue of variable-length sequence inputs, the sliding window inference technique demands substantial computational resources, resulting in slower processing speeds.

To further overcome the limitations associated with input length, Dai et al. [68] introduced Transformer-XL in 2018. This model enhances the standard Transformer by allowing for the acceptance of arbitrary-length sequence inputs while improving training efficiency. Transformer-XL achieves its objectives through two core mechanisms. First, it implements a segmented recursive mechanism, which divides long sequences into shorter segments for individual processing. This approach reduces the model's reliance on global sequence information and minimizes computational resource consumption. Second, Transformer-XL incorporates a relative position encoding mechanism, which enriches the model's capacity to learn sequence structures by integrating relative positional information within each segment. Specifically, this mechanism employs a unique embedding method to capture the relative distances and order relationships among elements in the input sequence, thereby enhancing the model's ability to comprehend complex patterns and improving its generalization capabilities. Consequently, the integration of segmented recursion and relative position encoding endows Transformer-XL with enhanced efficiency and accuracy in managing long sequence inputs, positioning it as a valuable asset in the fields of NLP and other domains that involve sequence data processing.

During the training phase, the Transformer-XL model utilizes an additional fixed-length storage space to preserve the hidden states of preceding segments. When processing the current segment, these stored hidden states interact with the current input, establishing a connection between the two segments. By introducing segmented recursion and relative position encoding, Transformer-XL effectively addresses the critical limitation of the standard Transformer model regarding arbitrary-length sequence inputs. The relative position encoding mechanism further facilitates the capture of distance information between distinct segments, thereby enhancing the model's comprehension of the overall sequence structure.

XLNet [69] builds upon the Transformer-XL architecture as a pre-training model designed to tackle the challenges associated with processing long text sequences. XLNet inherits the pivotal mechanisms of segmented recursion and relative position encoding from Transformer-XL, enabling it to accept lengthy sequences as inputs while exhibiting rapid reasoning capabilities. During the pre-training stage, XLNet is trained on five extensive corpora, with a cumulative size exceeding 150 GB, thus providing the model with a vast amount of data. These corpora include BooksCorpus, Wikipedia, Giga5, ClueWeb 2012-B, and Common Crawl. Additionally, XLNet introduces a permutation language model that leverages a permutation mechanism, synthesizing the strengths of both autoregressive and auto-encoding language models while mitigating their limitations. This approach allows for effective modeling of language context dependencies. Furthermore, through the mechanisms of segmented recursion and relative position encoding inherited from Transformer-XL, XLNet enhances its capability to process long texts. Overall, the XLNet model successfully models language context dependencies by employing permutation mechanisms alongside the segmented recursion and relative position encoding of Transformer-XL, thereby significantly improving performance in various NLP tasks.

### 2.4. BART

The BART model [70] employs a Transformer architecture comprising standard 6-layer encoders and 6-layer decoders. During the pre-training phase, BART utilizes document recovery as its target task, wherein the input document sequence is intentionally corrupted, and the decoder is tasked with reconstructing the original sequence. This "destruction" and "restoration" process effectively serves to denoise the data. To simulate real-world document noise conditions, BART incorporates five distinct corruption methods during pre-training: symbol masking, symbol deletion, text filling, sentence rearrangement, and document rotation. For fine-tuning and adapting to various downstream tasks, the BART model provides tailored architectures and methodologies to accommodate different task types, including sequence classification, token classification, sequence generation, and machine translation.

In contrast, the T5 model [71] aspires to establish a unified framework that treats a wide array of NLP tasks as text-to-text tasks. This approach enables the execution of all NLP tasks using a consistent model, loss function, training regimen, and decoding process, encompassing activities such as reading comprehension, summary generation, and text classification. To achieve this objective, the T5 model incorporates several enhancements, including the removal of the Layer Normalization bias, the relocation of Layer Normalization outside the residual path, and the implementation of a distinct position embedding. These modifications significantly enhance the model's generalization capabilities and robustness, allowing it to handle diverse NLP tasks with greater efficacy.

### 2.5. GPT

Transformer Decoder-based PLMs have demonstrated exceptional performance in language generation tasks, notably with models such as GPT achieving remarkable results across various NLP applications. GPT-2 capitalizes on larger datasets and model architectures for pre-training and introduces the innovative concept of zero-shot learning, wherein a pre-trained model can be directly applied to downstream tasks without the need for fine-tuning. GPT-3 [44] further enhances this framework by increasing the parameter count to 175 billion and adopting a few-shot learning approach [72,73]. For each subtask, GPT-3 requires only 10 to 100 training samples, and in some instances, it can successfully complete tasks with as few as one training sample or even none. These features significantly enhance the model's generalization capabilities, enabling it to effectively address a diverse range of NLP tasks.

Moreover, InstructGPT [74], a PLM based on GPT-3, is fine-tuned using human feedback to better align with user intent across various tasks. This development led to the creation of ChatGPT, which is specifically designed for reasoning and dialogue tasks. Building on the foundational principles of InstructGPT, ChatGPT incorporates instructional learning during retraining. This process involves manual evaluation of the responses generated by ChatGPT, followed by adjustments to the model parameters to facilitate the generation of answers that more closely align with human cognitive patterns. However, it is noteworthy that ChatGPT currently lacks the capability to utilize text classification datasets for direct fine-tuning. Researchers can only employ ChatGPT for a limited number of text classification tasks, and its performance in these tasks is generally inferior to that of other fine-tuned models based on open-source PLMs [75].

GPT-4 employs the same training methodology as GPT-3 to train ChatGPT, enabling it to handle multi-modal data. Following retraining, ChatGPT outperforms GPT-3 in Q&A tasks. With the development of the GPT series of models, Prompt Learning has emerged as a prominent research area [76–78]. Prompt Learning aims to facilitate large models in recalling the knowledge they acquired during the pre-training phase.

For an input text $x$, a function $f_{prompt}(\cdot)$ exists that transforms $x$ into prompt form $\acute{x}$.

$$\acute{x} = f_{prompt}(x) \tag{3}$$

The inclusion of prompts at the beginning or end of a sentence can significantly influence the model's output.

In recent years, the increasing impact of the GPT series on the academic community has led to a continuous expansion in the scale of various PLMs. Numerous research initiatives are actively investigating the upper limits of model parameter sizes. For instance, the Gopher model [79] comprises 280 billion parameters, the Megatron-Turing NLG model [80] boasts 530 billion parameters, and the PaLM Chowdhery model [81] features 540 billion parameters. The emergence of these large-scale PLM models presents both new opportunities and challenges within the field of NLP.

## 3. Text classification method based on deep learning

The automated processing of diverse textual data is a fundamental aspect of NLP, with text classification serving as a critical task within this domain. Traditional methods for text classification can be broadly categorized into two groups: machine learning-based methods and deep learning-based methods. Conventional machine learning approaches often necessitate complex feature selection and extraction processes, and they frequently lack the capacity for robust feature detection. This limitation can result in suboptimal classification performance.

Early research in text classification employed various methodologies, including but not limited to support vector machines [82], naive Bayes [83], decision trees [84], logistic regression [85], and term frequency-inverse document frequency (TF-IDF) [86]. While these approaches were widely utilized, deep learning-based methods have emerged as a dominant paradigm due to their superior capabilities in feature detection and extraction, which arise from the adaptive nature of neural networks when processing textual data.

In contrast to traditional machine learning techniques, deep learning-based approaches leverage neural networks to automatically extract features from raw textual data, thereby eliminating the need for manual feature engineering. This shift has led to enhanced performance across various text classification tasks [87,88], resulting in deep learning methods gradually supplanting traditional machine learning techniques in this field.

Deep learning-based methods employ deep neural networks to perform multi-level feature extraction and learning on textual data. These network architectures excel at capturing complex patterns and semantic information inherent in text [89–91]. In the context of applying deep learning to text classification, the preprocessing step involving word vector representation is particularly crucial. Word vector representation is a numerical encoding technique that facilitates the representation of textual data for computational processing.

Traditional word vector representations, such as one-hot encoding, often fail to effectively capture the spatial relationships between different words due to their lack of similarity information. To address this challenge, the Google team introduced a word vector representation tool called Word2Vec [92] in 2013. This method, trained on Google News data, is capable of capturing distance relationships between words more effectively, making it widely adopted in various NLP tasks. Another prominent tool for word vector representation is GloVe [93].

The integration of these tools has become a foundational element for NLP tasks. In most existing deep learning-based text classification methods, CNNs or RNNs are commonly employed for classification tasks [94,95]. However, the performance of deep learning algorithms is heavily contingent upon the quality of input features [96]. Consequently, significant research efforts have been devoted to the design of effective features [97,98] or the learning of expressive features from neural networks. Among these methods, Word2Vec and GloVe are often utilized as pre-training tools for neural networks [99–101], enabling the capture of distance relationships between features. Subsequently, various neural network architectures are employed to extract high-level features, forming the foundation for classification.

### 3.1. Text classification method based on CNNs

In 2014, Kim [10] introduced a text classification model based on CNNs, which demonstrated promising results in various text classification tasks, thereby validating the potential of deep learning approaches in this domain. Deep learning-based text classification models can be categorized into shallow neural networks [10,102], complex deep neural networks [103], and other variations, depending on the specific requirements of the task. When trained on substantial datasets, these models can significantly enhance the performance of text classification tasks. Fig. 6 illustrates a standard framework for sentence classification utilizing CNNs.

The process of extracting features using CNNs involves two steps: the first step is to apply convolution operations to extract higher-level local features, and the second step is to aggregate these local features using max pooling. This reduces the complexity of the network while retaining the most significant and important text features. Each word in a sample is represented as $i_k \in R^m$, which denotes the m-dimensional word vector of the $k$th word in a sentence. A sentence $T$ is represented as shown in Eq. (4), where $T$ contains $k$ words.

$$T = i_1, i_2, \ldots, i_k \tag{4}$$

Let $\mathbf{w}_n \in R^{hk}$ represent the convolution filter used to convolve the word vector matrix. The feature $y_n$ is the new text feature obtained by convolving the sentence $T$ using the convolution filter $\mathbf{w}_n$, as shown in Eq. (5).

$$y_n = f(\mathbf{w}_n \cdot T + b) \tag{5}$$

Here, $b \in R$ is a bias term, and $f(\cdot)$ represents an activation function. By convolving the sentence $T$ with filters of different sizes, multiple local features can be obtained, forming a set of feature maps $Y$ as shown in Eq. (6).

$$Y = [y_1, y_2, \ldots, y_n] \tag{6}$$

To reduce network complexity while retaining significant features, max pooling operations are applied to the feature maps to find the maximum value among these features, as shown in Eq. (7).

$$z = max(Y) \tag{7}$$

The pooling feature maps form the 6th layer of the network and are passed to the output layer using a fully connected network. After training and optimizing the network parameters, the probability distribution of the output classes is obtained. In the output layer, a classifier is built using a softmax function because its output represents a conditional probability distribution, as shown in Eq. (8).

$$softmax_j = \frac{\exp(z)}{\sum_{j=1}^{C} \exp(y_j)} \tag{8}$$

Here, $C$ represents the class, $y$ is the input of the fully connected layer, which corresponds to the pooling feature maps.

The outlined process elucidates the application of CNNs for text classification, underscoring the successful implementation of deep learning in this area. For instance, Tang et al. [104] achieved commendable results in sentiment classification using CNNs while considering both user preferences and overall product quality. Some researchers have sought to incorporate word order information into shallow neural networks to enhance classification performance [102], while others have employed deeper neural networks for text classification. Notably, Zhang et al. [103] proposed a nine-layer neural network featuring up to

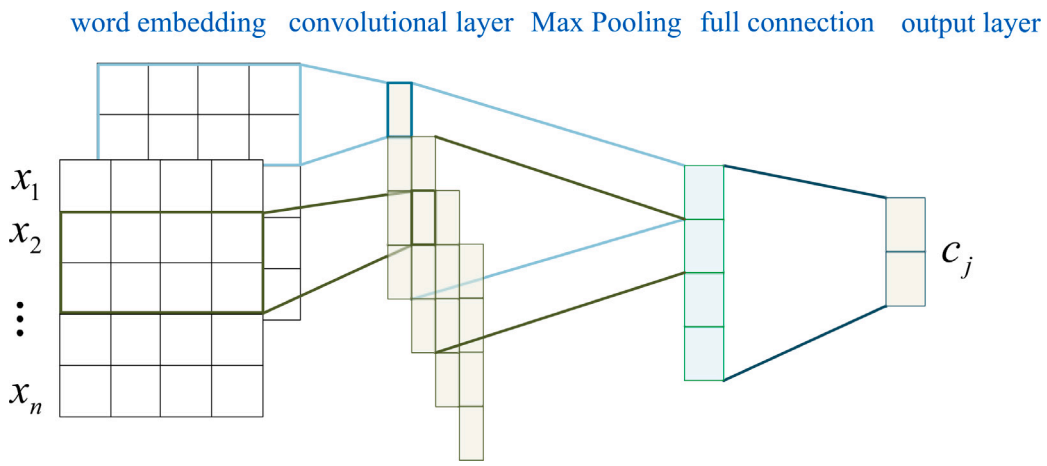


**Fig. 6.** A typical model framework for sentence classification based on CNNs.

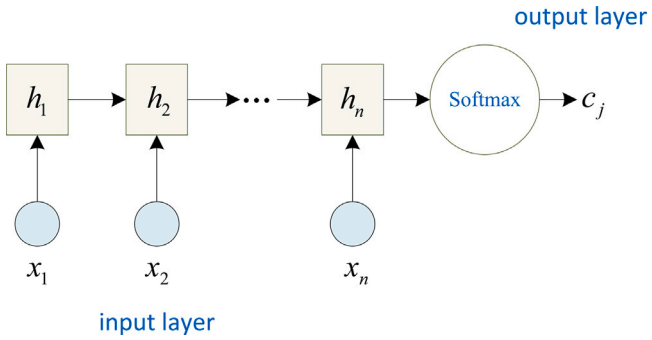


**Fig. 7.** A typical example of a text classification method based on RNNs.

six convolutional layers for feature extraction and three fully connected layers, specifically designed to process characters rather than words.

The character-level CNN [103] approach extracts local text features through the utilization of up to six convolutional layers, without necessitating knowledge of words or any linguistic or semantic structure. This methodology has demonstrated effective performance on large-scale text datasets. In 2016, Kim et al. [105] proposed a hybrid method that integrates convolutional neural networks and long short-term memory (LSTM) networks to extract character-level text features, which can be applied to text classification tasks across multiple languages. Additionally, various research outcomes have provided novel solutions and insights for character-level text classification tasks [106, 107].

### 3.2. Text classification method based on RNNs

RNNs have become widely adopted for text classification within the realm of deep learning [108]. These models excel at extracting relevant features by leveraging contextual information present in the text, making them particularly effective for handling long text datasets. Fig. 7 illustrates a typical framework for text classification based on RNNs.

The Recurrent Convolutional Neural Network (RCNN) [109] represents a novel architecture that amalgamates the strengths of both CNNs and RNNs. It employs a recurrent structure to capture contextual information from text sequences while utilizing max pooling techniques to aggregate essential features. The RCNN not only enhances the accuracy of text classification but also reduces model complexity, thereby providing an effective solution for various text classification tasks and underscoring the pivotal role of RCNNs in deep learning-based text classification.

Additionally, Yang et al. [110] introduced a text classification model that incorporates an attention mechanism, which elucidates the significance of each text feature within the current document. This approach has been shown to enhance the accuracy of text classification to a notable extent. In 2017, Qin et al. [111] proposed a method utilizing Generative Adversarial Networks (GANs) to tackle the implicit text relation classification problem. Building upon this foundation, Liu et al. [112] developed a GAN-based adversarial multi-task learning framework aimed at addressing text classification challenges. Due to the inherent characteristics of GANs, this framework demonstrates strong performance in semi-supervised learning scenarios, effectively improving the accuracy of text classification, particularly in cases involving small sample sizes or insufficient labels [113].

Since 2019, Graph Neural Networks (GNNs) have been increasingly applied to text classification tasks. Yao et al. [114] presented a text classification model based on GNNs, followed by Liu et al. [115], who proposed a tensor graph convolutional network specifically designed for text classification. To overcome the limitations of existing GNNs in capturing contextual information, Zhang et al. [116] introduced a graph-based inductive text classification method. This innovative approach constructs a separate graph document for each text instance and employs GNNs to learn fine-grained local word representation structures.

### 3.3. Text classification method based on capsule network

The concept of capsule networks was first introduced by Sabour et al. [117] in 2017. In 2018, Wang et al. [118] proposed an innovative method for emotion classification that integrates RNNs with capsule networks to enhance classification accuracy. This approach capitalizes on the robust sequence modeling capabilities of RNNs and the advanced feature representation abilities of capsule networks, thereby offering a promising solution for sentiment classification tasks. Fig. 8 illustrates a typical methodology based on capsule networks for text classification.

In the capsule network, a crucial step is to implement a routing algorithm to enable communication between two capsule layers. Specifically, this is achieved by facilitating information transfer between the PrimaryCaps layer and the DigitCaps layer to replace the fully connected layer. The length of the DigitCaps layer ($L_2$ norm distance) represents the probability of class existence. The routing algorithm aims to determine an optimal coefficient $A_{ij}$ to achieve the best information transmission effect. Initially, the coefficient $A_{ij}$ is set to $\frac{1}{k}$, which signifies that the next layer capsule is a weighted sum of each capsule in the previous layer, with equal initial weights. The goal of the routing algorithm is to discover the optimal weight coefficient.

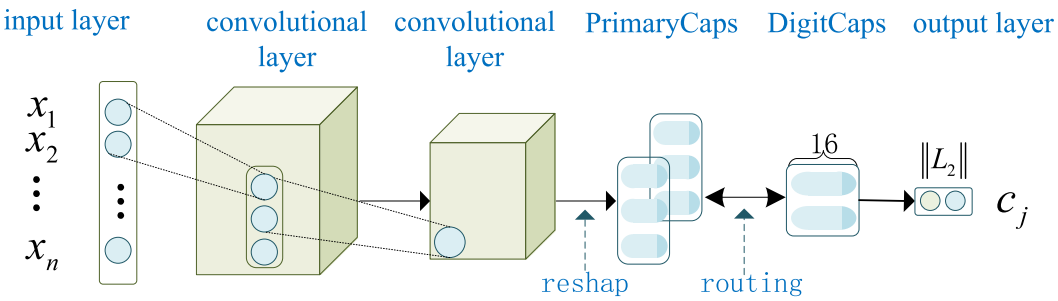


**Fig. 8.** A typical method based on the capsule network for text classification.

Firstly, the coefficient $A_{ij}$ is obtained through a variable $B_{ij}$, and its calculation process is determined by Eq. (9), where the initial value of $B_{ij}$ is 0.

$$A_{ij} = \frac{exp(B_{ij})}{\sum_k exp(B_{ik})} \tag{9}$$

After obtaining the coefficient $A_{ij}$, an intermediate variable $s_j$ is calculated using Eq. (10).

$$s_j = \sum_i A_{ij} \cdot W_{ij} \cdot u_i \tag{10}$$

Following this, $v_j$ in the digital capsule layer is calculated using Eq. (11).

$$v_j = sj21 + sj2sjsj \tag{11}$$

Finally, the new coefficient $B_{ij}$ is calculated using Eq. (12), completing one routing iteration process.

$$B_{ij} = B_{ij} + W_{ij} \cdot u_i \cdot v_j \tag{12}$$

Here, $W_{ij}$ is a fixed shared weight matrix. Typically, after about three routing iterations, the capsule network attains optimal performance.

The final layer of the capsule network is the class layer, which utilizes the length of the capsule to indicate the probability of each class. The length of the output of the digital capsule layer represents the probability of the existence of a particular class, and the coefficient $A_{ij}$ is updated through a routing algorithm, while other parameters and shared weight matrix $W_{ij}$ are updated based on the loss function.

Yang et al. [119,120] investigated the capsule network model with a dynamic routing mechanism, presenting a novel solution for text classification tasks. By incorporating this dynamic routing mechanism, the model is better equipped to capture contextual information within the text, thereby enhancing classification accuracy. Furthermore, in 2019, Zhao et al. [121] proposed a scalable and reliable capsule network model that exhibits strong adaptability to various data types and tasks, effectively supporting applications such as multi-label text classification and intelligent question answering.

In 2019, Chen et al. [122] introduced a transfer capsule network model that adeptly addressed the challenges associated with aspect-level sentiment classification by applying document-level knowledge transfer to sentiment analysis tasks. This model provided new insights for advancements in this field. In 2020, Wu et al. [11,14,15] presented a word-level capsule network for text classification, achieving significant progress in the application of capsule networks to natural language processing tasks.

## 4. Text classification method based on PLMs

Previous research predominantly utilized Word2Vec [92,101] or GloVe [93] as pre-training tools for text classification. However, since 2018, word-level text classification models based on BERT have gradually emerged as the dominant approach. BERT [29], a PLM developed by Google, has gained prominence due to advancements in self-supervised learning techniques, positioning PLMs as critical tools for visual and linguistic representation learning [123–126].

The process of employing the parameters of a PLM as initial parameters for related tasks is referred to as pre-training. By pre-training the model on large-scale unlabeled datasets and subsequently fine-tuning it on labeled data for specific tasks, the model can extract a wide array of common features, thereby alleviating the learning burden associated with specific tasks. Currently, fine-tuning PLMs for precise classification in text classification tasks has become a widely adopted methodology.

### 4.1. Text classification method based on fine-tuning

In recent years, PLMs have demonstrated substantial potential in text classification tasks. Notably, PLMs, particularly BERT, have been extensively applied across various NLP tasks, owing to the rich knowledge they accumulate during pre-training [127]. By constructing downstream task models based on PLMs as foundational models and fine-tuning these models, researchers can fully leverage the knowledge acquired from PLMs, thereby enhancing model performance [128,129].

Through self-supervised learning and the utilization of extensive corpora, PLMs acquire significant knowledge during the pre-training phase. For instance, BERT is founded on the Transformer Encoder architecture and employs English corpus data for self-supervised training, acquiring a considerable amount of semantic knowledge through masked language modeling and next sentence prediction tasks.

The construction of text classification models utilizing PLMs has emerged as a mainstream approach. Guo et al. [34] proposed a contrastive learning method to derive effective representations, which was applied to text classification based on BERT's single-language embedding and to Alzheimer's disease detection.

Chen et al. [130] enhanced model interpretability by visually representing different combinations of words and phrases within a hierarchical structure, thereby detecting the influence of features at varying levels. Croce et al. [131] introduced a semi-supervised generative adversarial network text classification method leveraging the BERT framework, wherein unlabeled data was employed to fine-tune the BERT architecture in a generative adversarial environment. Qin et al. [132] presented a feature projection method based on BERT to project neutral features for high-precision classification, thereby improving the performance of BERT-based text classification models.

Fig. 9 illustrates the overall architecture of a text classification method based on PLMs. This method employs two text encoders to extract label features and text sequence features. Various pre-trained models, such as BERT [29], can be utilized to extract text sequence features, which are then compared to achieve classification.

Here, $S$ represents the text features extracted by the Encoder, which takes the text sequence $D$ as input and obtains the text feature representation through the following function:

$$S = Encoder_S\left(D * w^T\right) \tag{13}$$

In the above formula, $w$ represents the operation of obtaining word embeddings. Typically, we use the last hidden state of the Encoder to represent text features. Firstly, the input sequence $D$ is mapped to word

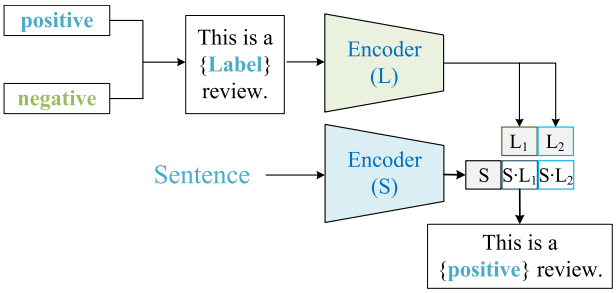


**Fig. 9.** A text classification method based on PLMs.

embeddings $E$, which are multiplied by the input sequence $D$ through a matrix $w$.

$$E = D * w^T \tag{14}$$

Then, the word embeddings $E$ are passed to the Encoder to obtain features $S$ with contextual correlation information. For each different label $C_j(j = 1, 2, \ldots, k)$, a prompt template is used to obtain a set of sentences $H_j(j = 1, 2, \ldots, k)$.

$$H_j = M\left(C_j\right) \tag{15}$$

Each sentence $H_j$ representing a class label is input into a text encoder to obtain the feature representation of the label as follows:

$$L_j = Encoder_L\left(H_j\right) \tag{16}$$

Then, the distances between $S$ and $L_j(j = 1, 2, \ldots, k)$ are calculated, and the minimum distance is selected as the final classification probability.

Despite the excellent performance of PLMs in obtaining feature representations, certain valuable information embedded within the labels has not been fully utilized [133]. To address this limitation, several research efforts have focused on extracting semantic information from labels and employing it as a data augmentation strategy [134].

Utilizing label information for data augmentation has proven to be effective [135]. Mekala et al. [136] successfully developed a contextualized corpus using BERT for generating pseudo-labels, thereby achieving a context-based weakly supervised text classification method. Giovanni et al. [137] strategically harnessed the semantic information of labels in text and classification tasks by generating labels during the prediction process. Hu et al. [138] expanded the label space by leveraging external knowledge bases and refined this expanded label space using PLMs before applying it for prediction, significantly enhancing the performance of zero-shot and few-shot text classification tasks.

Chen et al. [139] proposed a label-aware data augmentation method based on dual contrastive learning for text classification tasks. This method treats labels as enhanced samples and employs contrastive learning to discern the correlation between input samples and enhanced samples. Mueller et al. [140] introduced a label semantics-aware pre-training model that utilizes labels to improve the generalization ability and computational efficiency of few-shot text classification. Guo et al. [141] developed an autoencoder called ZeroAE, which encodes two independent spaces based on BERT encoders — namely, label-related space (for classification) and label-unrelated space — and subsequently decodes these latent spaces using GPT-2 to recover text and generate labeled text in unseen domains for encoder training.

Yang et al. [142] proposed a prototype-guided semi-supervised model that integrates a prototype anchoring comparison strategy with a prototype-guided pseudo-label strategy. Both strategies cluster similar texts, achieving a high-density distribution of analogous texts, thereby alleviating the issue of decision boundary misfit. Lee et al. [143] utilized question-answering datasets to facilitate data augmentation for text classification within the educational domain. Clarke et al. [144] introduced two pre-training strategies — implicit and explicit pre-training
— to enhance the generalization capability of PLMs in text classification tasks.

### 4.2. Text classification method based on prompts

Text classification is a fundamental research area within the field of NLP. With the advent of PLMs, many research efforts have increasingly focused on leveraging these models to perform text classification tasks. The construction of text classification models utilizing PLMs has become the predominant approach, falling under the umbrella of transfer learning, which allows for the completion of text classification tasks with a limited number of samples [145] . However, due to constraints related to computational resources and the scale of PLM models, some studies have opted not to fine-tune PLMs for text classification. Instead, they have adopted zero-shot or few-shot text classification methodologies based on prompt learning.

Wang et al. [146] proposed a unified prompt tuning framework that significantly enhances the performance of BERT-style few-shot text classification by explicitly capturing prompt semantics from non-target NLP datasets. Zhang et al. [147] innovatively integrated image feature information into sentence modeling for text classification, thereby improving performance through the effective incorporation of visual features. Additionally, Zhang et al. [148] introduced a meta-learning framework that simulates a zero-shot learning scenario by utilizing existing classes alongside virtually non-existent classes, offering a novel solution for zero-shot learning. Gera et al. [149] presented a plug-and-play method that employs self-training to facilitate learning for the target task, thereby proposing a new approach to model training. Furthermore, Zhang et al. [150] developed a random text generation technique that produces high-quality contrastive samples, significantly enhancing the accuracy of zero-shot text classification and providing a fresh perspective on zero-shot learning.

Few-shot text classification involves training a model with only a small number of samples, with performance improvement achieved through prompt fine-tuning. This area has garnered considerable attention, as researchers continuously explore innovative methods to enhance classification performance in small-sample scenarios. Nishikawa et al. [151] proposed a multi-language entity bag model (Bag-of-Entities) that effectively improves zero-shot cross-language text classification performance by extending multi-language PLMs. This research introduces a new approach to cross-language text classification, addressing inherent challenges associated with such tasks. Min et al. [152] introduced a noise channel method to adjust language model prompts and update model parameters in a constrained manner, thereby achieving low-sample text classification. By accounting for the influence of noise during parameter adjustments, this method enhances the robustness and classification performance of the model. Zha et al. [153] proposed a self-supervised hierarchical task clustering method that strengthens the relationships between tasks by dynamically learning knowledge from different clusters, thereby improving interpretability. This approach offers insights into the decision-making processes of the model, enhancing both interpretability and reliability. Zhao et al. [154] introduced an explicit and implicit consistency regularization-enhanced language model that improves generalization ability through these regularization techniques, resulting in increased accuracy for few-shot text classification. Collectively, these research efforts present innovative ideas and methodologies for small-sample text classification, contributing to enhanced classification performance and generalization ability, and holding significant theoretical and practical implications.

Zhang et al. [155] proposed a prompt-based meta-learning model that achieves robust few-shot text classification by delegating the label

word learning task to a base learner while assigning the template learning task to a meta-learner. Shnarch et al. [156] introduced an intermediate unsupervised classification task situated between the pre-training and fine-tuning stages to bolster model performance, thereby advancing the field of unsupervised learning in small-sample contexts. Zhao et al. [157] presented a memory-mimicking meta-learning method that improves model performance by enhancing reliance on task-adapted support sets, effectively boosting performance across various tasks. To address the issue of insufficient representation in small validation sets within low-resource settings, Choi et al. [158] proposed an early stopping method that utilizes unlabeled samples to better estimate the class distribution, thereby improving model performance on unlabeled data.

Wang et al. [159] developed a framework that amalgamates conceptual knowledge through keyword extraction based on prompts, weight allocation for each prompt keyword, and final representation within a knowledge graph embedding space for text classification in extreme zero-shot settings. Qin et al. [160] introduced a novel zero-shot text classification method that reformulates sample header text classification as a text-image matching problem, applicable via CLIP [161]. Wang et al. [162] proposed a contrastive learning framework designed to enhance zero-shot text classification performance by integrating prompt templates with tags and learning the distance between these tags and sentences. Liu et al. [163] addressed zero-shot text classification by utilizing unlabeled data and adjusting the language model, proposing a new learning objective termed first sentence prediction to bridge the gap between unlabeled data and text classification tasks. Yu et al. [164] introduced a retrieval-augmented framework for generating training data from unlabeled corpora in general domains, which can be leveraged to enhance zero-shot text classification performance. These research initiatives present innovative strategies and methodologies for zero-shot text classification, contributing to improved classification outcomes and holding substantial theoretical and practical significance.

## 5. Widely used datasets and evaluation metrics

### 5.1. Widely used datasets

In the realm of text classification, the availability and quality of annotated datasets have emerged as crucial factors propelling the rapid advancement of this research domain. These datasets exhibit notable features, such as domain coverage, class variety, text length, and dataset magnitude, which play a significant role in determining the outcomes of text classification experiments. Table 1 presents a comprehensive outline of these datasets, encompassing the number of classes incorporated in each dataset, the mean sentence length, and the dataset's size. Therefore, this survey provides a detailed overview of commonly utilized open text classification datasets, emphasizing their relevance and significance in the field of text classification.

**AG News** [103] is a news text corpus designed for the academic community. It comprises of 127,600 news articles, with each sample labeled by a concise text and accompanied by four class labels. The input for each news item is derived from its title and description text.

**Amazon Reviews** [103] extracted from Amazon customer reviews and star ratings, is widely employed in diverse research studies. This dataset consists of five categories, with the complete corpus containing 600,000 training samples and 130,000 test samples per class. This version is referred to as Amazon Reviews.F. Another variation, Amazon Reviews.P, encompasses data for two categories with 3,600,000 training samples and 400,000 test samples, catering to distinct research requirements.

**CR** [165] is a customer review corpus where each sample is labeled as positive or negative. It comprises of a total of 3770 samples.

**Dbpedia** [166] is an information-rich corpus derived from Wikipedia, utilizing community crowdsourcing. It encompasses data for a comprehensive range of 14 distinct categories. Within each class, we have randomly selected 40,000 training samples and 5000 test samples. Consequently, the dataset comprises a total of 560,000 training samples and 70,000 test samples.

**IMDb** [167] is a dataset specifically curated for NLP and text analysis, comprising 50,000 movie reviews. This dataset is divided into two sets of equal size, with 25,000 samples allocated for training and another 25,000 samples for testing. It is widely recognized as one of the most frequently employed datasets in the realm of text classification applications.

**MPQA** [168] is a sentiment classification dataset extensively employed in tasks related to emotional classification. It consists of 10,604 samples extracted from news articles originating from various sources. The dataset contains two class labels, namely positive and negative, with 3311 positive samples and 7293 negative samples.

**MR** [169] is a dataset comprising user reviews for diverse movies available on the web. Each review is labeled with a sentiment score of either positive or negative. The dataset encompasses a total of 5331 positive samples and an equal number of negative samples.

**Sogou News** [103] is a comprehensive corpus derived from the Sogou News database. The news articles in this dataset are classified into five distinct categories based on the domain name in the URL. The corpus comprises a total of 510,000 samples, encompassing five categories of news articles. Each class includes 90,000 training samples and 12,000 test samples.

**SST** [99] is a specialized dataset curated for movie review sentiment classification. It consists of comments posted by movie enthusiasts on the internet and is divided into five categories based on the intensity of sentence sentiment: very positive, positive, neutral, negative, and very negative. The dataset encompasses a total of 11,855 texts and is referred to as SST-1. Additionally, another version of SST-2 adopts binary labels to classify reviews into positive and negative sentiment groups, comprising 9613 binned texts.

**SUBJ** [170] is a dataset designed for classifying user comments as either subjective or objective. Each sample usually constitutes a sentence and comprises a total of 9999 samples. These sentences are classified into two categories: subjective and objective.

**TREC** [171] is a corpus extracted from the TREC question dataset. The sentences in this dataset are classified based on question type, encompassing a total of 5891 samples.

**Yahoo Answers** [172] is a dataset that comprises ten distinct categories of data, all extracted from Yahoo Answers. Each class includes 140,000 training samples and 6000 test samples.

**Yelp Review** [103] is derived from the review comments of Yelp's 2015 challenge dataset. This dataset encompasses five distinct categories, with each review classified according to scores that range from one to five stars, referred to as Yelp Revie.F. Each class contains 130,000 training samples and 10,000 test samples. Additionally, the Yelp Review.P dataset consists of 280,000 training samples and 19,000 test samples, organized into two classes.

**20Newsgroups** [109] dataset is a widely used collection of English news texts, primarily utilized for news classification research. This dataset is derived from the publication and collection of newsgroup documents on 20 different topics, each containing a certain number of samples, collectively constituting 18,846 unique texts. The dataset covers a diverse range of topics, including politics, religion, sports, and technology, among others.

The characteristics of the dataset, including the number of classes (class), average sentence length (Avglength), and total size (Size), are presented in Table 1. These attributes have been previously discussed and elucidated to provide a comprehensive understanding of the dataset's properties and features. Specifically, the number of categories in Table 1 denotes the count of distinct categories within each dataset, the average length signifies the mean sentence length across all categories, and the total size represents the overall magnitude of the dataset. The detailed information pertaining to these

**Table 1**
Summary statistics for the datasets.

| Dataset | Class | Avg.L | Size | Reference |
|---|---|---|---|---|
| AG News | 4 | 16 619 | 127,600 | [103] |
| Amazon Review.F | 5 | 93 | 3,600,000 | [103] |
| Amazon Review.P | 2 | 91 | 4,000,000 | [103] |
| CR | 2 | 19 | 3775 | [165] |
| Dbpedia | 14 | 55 | 630,000 | [166] |
| IMDb | 2 | 294 | 50,000 | [167] |
| MPQA | 2 | 3 | 10,604 | [168] |
| MR | 2 | 20 | 10,662 | [169] |
| Sogou News | 5 | 578 | 510,000 | [103] |
| SST1 | 5 | 18 | 11,855 | [99] |
| SST2 | 2 | 19 | 9613 | [99] |
| SUBJ | 2 | 23 | 9999 | [170] |
| TREC | 6 | 10 | 5891 | [171] |
| Yahoo Answers | 10 | 112 | 146,000 | [172] |
| Yelp Review.F | 5 | 155 | 131,000 | [103] |
| Yelp Review.P | 2 | 153 | 299,000 | [103] |
| 20Newsgroups | 20 | 221 | 18,846 | [109] |

attributes serves to enhance our comprehension of the dataset's structure and characteristics, enabling us to effectively leverage the data for subsequent research and analysis. By considering these attributes, researchers can gain valuable insights into the dataset's composition and make informed decisions regarding its applicability to specific research objectives.

### 5.2. Evaluation metrics

When evaluating the performance of text classification models, common evaluation metrics include accuracy (Accuracy), error rate (Error Rate), precision (Precision), recall (Recall), and F1 score (F1 Score). These evaluation metrics are calculated based on the following classification results:

TP: The number of samples correctly classified as positive samples. FP: The number of samples incorrectly classified as positive samples, i.e., the number of negative samples falsely reported as positive. TN: The number of samples correctly classified as negative samples. FN: The number of samples incorrectly classified as negative samples, i.e., the number of positive samples missed. The calculation expression for accuracy is:

$$Accuracy = \frac{TP + TN}{N} \tag{17}$$

The calculation expression for error rate is:

$$ErrorRate = 1 - Accuracy = \frac{FP + FN}{N} \tag{18}$$

Precision reflects the proportion of samples correctly predicted and classified among all samples predicted as positive, and its calculation expression is:

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

Recall reflects the proportion of samples correctly predicted and classified among all samples actually classified as positive, and its calculation expression is:

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

Finally, F1 score is the harmonic mean of precision and recall, which is used to comprehensively consider both metrics, and its calculation expression is:

$$F1\ Score = \frac{2Precision * Recall}{Precision + Recall} \tag{21}$$

Generally, when the $F1$ score is high, the experimental method can be considered effective. Generally, a high F1 score indicates an effective experimental method. Therefore, researchers should consider these evaluation metrics when assessing the performance of text classification models to ensure accurate and reliable results.

### 5.3. Performance of commonly used PLMs on significant datasets

This study aims to conduct a comprehensive evaluation and comparative analysis of the performance of various pre-trained models across multiple datasets. For this purpose, nine widely utilized pre-trained models were selected for experimentation: BART-base (abbreviated as BART) [70], BERT-base (abbreviated as BERT) [29], DeBERTa-base (abbreviated as DeBE) [173], ELECTRA-base (abbreviated as ELEC) [64], Longformer-base (abbreviated as Long) [174], MPNet (abbreviated as MP) [65], Muppet-RoBERTa-base (abbreviated as Mup) [175], RoBERTa-base (abbreviated as RoBE) [176], and XLNet-base (abbreviated as XL) [8]. In the design of the output layer for each model, a linear layer was incorporated following the final classification (CLS) token.

To ensure a fair comparison, the study employed the base versions of each model, avoiding the use of their smaller or larger variants. The input sentence length for all models was standardized to 768 tokens. Experimental parameter settings were uniformly established, with a batch size of 16, a total of 20 iterations, a test set proportion of 10%, and a learning rate set to 1e-05.

Regarding the experimental hardware configuration, adjustments were made based on the dataset size. For large datasets, such as Amazon Review.F, Amazon Review.P, 20 Newsgroups, Dbpedia, Sogou News, Yahoo Answers, Yelp Review.F, and Yelp Review.P, computations were conducted using one A100 GPU to accommodate the high video memory requirements. Conversely, for smaller datasets, including CR, IMDB, MPQA, MR, SST1, SST2, SUBJ, and TREC, two GV100 GPU were utilized to expedite calculations. The experimental results are summarized in Table 2, which presents the performance of each pre-trained model across different datasets.

As illustrated in Table 2, the Muppet-RoBERTa model achieved the highest average accuracy, attributable to its advanced pre-training methodology that significantly enhances model performance. The XL-Net model also demonstrated commendable performance, particularly in processing long text sequences. In contrast, earlier models such as BERT exhibited slightly lower accuracy, primarily due to the relatively smaller training sample sizes utilized during their initial training phases compared to more recent models. Overall, a positive correlation was observed between model performance and scale; larger models generally outperformed their smaller counterparts. Furthermore, an increase in training data was found to effectively enhance model performance.

## 6. Future research challenges and trends

### 6.1. Future research challenges

The field of text classification faces several future challenges that necessitate ongoing research and innovation. These challenges encompass various dimensions, including:

(1) **Addressing More Complex Datasets**: As application scenarios expand in complexity, there is an increasing demand to confront more challenging datasets. Such datasets may involve tasks like multi-step reasoning questions and text classification for multilingual documents. Effectively addressing these challenges requires the development of sophisticated models and algorithms capable of managing their intricacies, thereby advancing research in this domain.

**Table 2**
Performance of commonly used pre-trained models on significant datasets.

| Dataset | BART | BERT | DeBE | ELEC | Long | MP | Mup | RoBE | XL |
|---|---|---|---|---|---|---|---|---|---|
| 20Newsgroups | 91.20 | 90.20 | 92.50 | 93.10 | 91.80 | 93.00 | 94.18 | 89.50 | 92.70 |
| AG_News | 95.20 | 95.05 | 95.44 | 95.20 | 94.99 | 95.62 | 96.12 | 95.68 | 95.55 |
| Amazon Review.F | 66.10 | 65.19 | 65.83 | 65.88 | 65.34 | 65.92 | 67.46 | 66.02 | 66.82 |
| Amazon Review.P | 97.37 | 96.68 | 97.36 | 97.42 | 96.49 | 97.92 | 98.26 | 97.84 | 98.06 |
| CR | 99.26 | 99.85 | 99.12 | 99.12 | 98.53 | 99.12 | 99.85 | 99.12 | 99.85 |
| Dbpedia | 99.32 | 99.20 | 99.36 | 99.16 | 99.06 | 99.18 | 99.42 | 99.26 | 99.38 |
| IMDB | 95.79 | 95.40 | 95.79 | 95.80 | 96.00 | 97.10 | 96.20 | 96.60 | 96.80 |
| MPQA | 92.83 | 91.98 | 93.68 | 93.02 | 91.23 | 92.74 | 92.83 | 90.94 | 93.11 |
| MR | 89.31 | 89.21 | 89.87 | 91.37 | 91.09 | 91.56 | 95.22 | 90.38 | 89.59 |
| Sogou News | 98.24 | 98.07 | 98.31 | 98.41 | 98.23 | 98.52 | 98.72 | 98.36 | 98.64 |
| SST1 | 52.41 | 52.49 | 57.22 | 55.19 | 54.60 | 57.13 | 61.69 | 57.22 | 61.94 |
| SST2 | 95.53 | 93.44 | 94.17 | 95.94 | 95.01 | 95.11 | 97.50 | 95.63 | 94.80 |
| SUBJ | 96.50 | 96.50 | 97.60 | 97.70 | 98.00 | 97.60 | 96.90 | 97.60 | 97.70 |
| TREC | 95.59 | 95.25 | 96.43 | 96.77 | 95.25 | 96.60 | 95.93 | 96.26 | 97.11 |
| Yahoo Answers | 77.62 | 76.26 | 77.92 | 78.12 | 77.98 | 78.62 | 79.02 | 78.64 | 78.82 |
| Yelp Review.F | 70.68 | 70.02 | 71.38 | 71.82 | 69.98 | 72.14 | 73.26 | 71.56 | 72.95 |
| Yelp Review.P | 98.11 | 97.84 | 98.19 | 98.14 | 97.10 | 98.32 | 98.50 | 97.90 | 98.63 |

(2) **Modeling Common Sense Knowledge**: The integration of common sense knowledge into PLMs has the potential to enhance model performance and generalization capabilities, particularly in contexts where information is incomplete. Future research should focus on effectively modeling and utilizing common sense within the PLM framework to better align with human knowledge. This exploration will facilitate the creation of text classification models that can more comprehensively capture the nuances and complexities of human language.
(3) **Multi-Task Learning and Cross-Domain Adaptation**: Many real-world text classification tasks require the simultaneous handling of multiple tasks or cross-domain challenges. Future research should investigate the design of effective algorithms for multi-task learning and cross-domain adaptation, aiming to improve overall performance. These algorithms should leverage shared knowledge across tasks and domains while remaining adaptable to the specific characteristics inherent to each task and domain.
(4) **Zero-Shot and Few-Shot Learning**: In numerous application scenarios, there is a pressing need to address classes that have not been encountered previously or have only a limited number of available samples. To tackle this issue, future research should concentrate on developing zero-shot and few-shot learning algorithms that facilitate broader applications and enhanced performance. These algorithms should be capable of learning from a limited number of samples and generalizing to unseen classes, thereby reducing reliance on extensive data annotation and enabling more flexible and scalable text classification systems.
(5) **Semantic Understanding and Information Extraction**: Beyond the primary goal of classifying text, text classification necessitates a deeper understanding of the semantic aspects of text and the extraction of relevant information. Achieving more comprehensive and accurate text processing involves exploring novel approaches that integrate NLP with information extraction techniques. Future research should focus on developing methodologies that effectively combine these two domains to enhance the overall performance of text classification systems. By leveraging the synergistic potential of NLP and information extraction, researchers can pave the way for advanced text classification models capable of comprehending underlying semantics and extracting pertinent information with precision and robustness.

In summary, future challenges in the field of text classification include addressing complex datasets, incorporating common sense modeling, exploring multi-task learning and cross-domain adaptation, tackling zero-shot and few-shot learning scenarios, and advancing semantic understanding and information extraction techniques. Researchers must continue to explore and innovate to overcome these challenges and drive the ongoing development of text classification technology.

### 6.2. Future trends

The field of text classification is poised for significant advancement, characterized by several emerging trends that are likely to shape the trajectory of research in this domain. These trends include:

(1) **Model Complexity and Computational Resources**: The continuous advancement of computer hardware capabilities facilitates the development of increasingly complex models capable of processing larger datasets, thereby enhancing model accuracy. However, this trend also necessitates greater computational resources, which may prompt a shift towards more efficient models and optimized computational strategies.
(2) **Novel Models and Algorithms**: Future research is expected to focus on exploring innovative model architectures and training methodologies that integrate common sense knowledge to improve generalization capabilities. Additionally, there will be an emphasis on designing effective algorithms for multi-task learning and cross-domain adaptation, as well as leveraging NLP technologies to enhance semantic understanding.
(3) **Expansion of Application Areas**: Text classification technology is progressively being deployed across diverse domains, including recommendation systems, sentiment analysis, information extraction, and question-answering systems. As technological advancements continue, the scope of application for text classification will likely expand further.
(4) **Challenges in Data Annotation**: Data annotation remains a critical yet often bottlenecked step in the text classification process due to the high costs and time commitments involved. Future research must investigate more efficient data annotation methods and explore unsupervised learning techniques to mitigate these challenges.
(5) **Privacy and Ethical Considerations**: As the utilization of text classification technology becomes more widespread, concerns related to privacy and ethics will gain prominence. Future research should prioritize the development of text classification systems that safeguard user privacy, ensure algorithmic fairness and transparency, and address ethical implications.

In summary, the future development trends in the field of text classification are likely to encompass increased model complexity, efficient algorithms, broader application areas, effective data annotation strategies, and heightened attention to privacy and ethical considerations. Researchers must continue to innovate and explore new avenues of inquiry to propel the ongoing evolution of text classification technology.

## 7. Conclusion

This survey offers a comprehensive overview of representative algorithms within the realm of PLMs in the context of text classification. It provides an in-depth examination of the primary characteristics of these PLM-based text classification methods. These methodologies not only augment the existing theoretical framework of text classification but also present novel perspectives and technical insights that can inspire research in related fields, thereby demonstrating their extensive research potential. Despite the significant advancements achieved by researchers in this domain in recent times, several pivotal challenges still necessitate further exploration and breakthroughs in future research endeavors.

## CRediT authorship contribution statement

**Yujia Wu:** Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Jun Wan:** Writing – review & editing.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yujia wu reports financial support was provided by Sanda University. Yujia Wu reports a relationship with Sanda University that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was Sponsored by Natural Science Foundation of Shanghai (No. 22ZR1445000) and Shanghai Higher Education Association's 2024–2026 planned research project (No. 2QZD2431).

## Data availability

No data was used for the research described in the article.

## References

[1] L. Wang, P. Xu, X. Cao, M. Nappi, S. Wan, Label-aware attention network with multi-scale boosting for medical image segmentation, Expert Syst. Appl. 255 (2024) 124698.
[2] X. Chen, J. Ke, Y. Zhang, J. Gou, A. Shen, S. Wan, Multimodal distillation pre-training model for ultrasound dynamic images annotation, IEEE J. Biomed. Health Inf. (2024).
[3] J. Wan, H. Liu, Y. Wu, Z. Lai, W. Min, J. Liu, Precise facial landmark detection by dynamic semantic aggregation transformer, Pattern Recognit. 156 (2024) 110827.
[4] J. Wan, J. Li, J. Chang, Y. Wu, Y. Xiao, X. Li, H. Zheng, Face alignment by component adaptive mechanism, Neurocomputing 329 (2019) 227–236.
[5] Y. Wu, J. Li, V. Chen, J. Chang, Z. Ding, Z. Wang, Text classification using triplet capsule networks, in: 2020 International Joint Conference on Neural Networks, IJCNN, 2020, pp. 1–7.
[6] Y. Wu, J. Li, C. Song, J. Chang, Words in pairs neural networks for text classification, Chin. J. Electron. 29 (2020) 491–500.
[7] E.A. Sağbaş, A novel two-stage wrapper feature selection approach based on greedy search for text sentiment classification, Neurocomputing 590 (2024) 127729.
[8] D. Wu, Z. Wang, W. Zhao, XLNet-CNN-GRU dual-channel aspect-level review text sentiment classification method, Multimedia Tools Appl. 83 (2) (2024) 5871–5892.
[9] G. Lefebvre, H. Elghazel, T. Guillet, A. Aussem, M. Sonnati, A new sentence embedding framework for the education and professional training domain with application to hierarchical multi-label text classification, Data Knowl. Eng. 150 (2024) 102281.
[10] Y. Kim, Convolutional neural networks for sentence classification, in: Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751.
[11] Y. Wu, J. Li, J. Wu, J. Chang, Siamese capsule networks with global and local features for text classification, Neurocomputing 390 (2020) 88–98.
[12] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P.S. Yu, L. He, A survey on text classification: From traditional to deep learning, ACM Trans. Intell. Syst. Technol. 13 (2020) 1–41.
[13] Y. Wu, X. Zhang, G. Xiao, H. Ren, Fusion of root and affix information with pre-trained language models for text classification, in: International Conference on Intelligent Computing, Springer, 2024, pp. 488–498.
[14] Y. Wu, J. Wan, Word and character semantic fusion by pretrained language models for text classification, in: 2024 International Joint Conference on Neural Networks, IJCNN, IEEE, 2024, pp. 1–8.
[15] Y. Wu, X. Guo, K. Zhan, CharCaps: character-level text classification using capsule networks, in: International Conference on Intelligent Computing, Springer, 2023, pp. 187–198.
[16] D. Tsirmpas, I. Gkionis, G.T. Papadopoulos, I. Mademlis, Neural natural language processing for long texts: A survey on classification and summarization, Eng. Appl. Artif. Intell. 133 (2024) 108231.
[17] K. Wang, Y. Ding, S.C. Han, Graph neural networks for text classification: A survey, Artif. Intell. Rev. 57 (8) (2024) 190.
[18] H. Ming, W. Heyong, Filter feature selection methods for text classification: a review, Multimedia Tools Appl. 83 (1) (2024) 2053–2091.
[19] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M.A. Chenaghlu, J. Gao, Deep learning based text classification: A comprehensive review, ACM Comput. Surv. 54 (2020) 1–40.
[20] M. Hong, H. Wang, Feature selection based on long short term memory for text classification, Multimedia Tools Appl. 83 (15) (2024) 44333–44378.
[21] Z. Cai, H. Zhang, P. Zhan, X. Jia, Y. Yan, X. Song, B. Xie, Multi-schema prompting powered token-feature woven attention network for short text classification, Pattern Recognit. 156 (2024) 110782.
[22] K. Li, C. Kang, Deep feature extraction with tri-channel textual feature map for text classification, Pattern Recognit. Lett. 178 (2024) 49–54.
[23] T. Feng, L. Qu, Z. Li, H. Zhan, Y. Hua, G. Haffari, IMO: Greedy layer-wise sparse representation learning for out-of-distribution text classification with pre-trained models, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 2625–2639.
[24] G. Gokceoglu, D. Çavuşoğlu, E. Akbas, Ö. Dolcerocca, A multi-level multi-label text classification dataset of 19th century ottoman and Russian literary and critical texts, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 6585–6596.
[25] D. Jiao, Y. Liu, Z. Tang, D. Matter, J. Pfeffer, A. Anderson, SPIN: Sparsifying and integrating internal neurons in large language models for text classification, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 4666–4682.
[26] Y. Wu, X. Guo, Y. Wei, X. Chen, ParaNet: Parallel networks with pre-trained models for text classification, in: International Conference on Advanced Data Mining and Applications, Springer, 2023, pp. 121–135.
[27] S. Jamshidi, M. Mohammadi, S. Bagheri, H.E. Najafabadi, A. Rezvanian, M. Gheisari, M. Ghaderzadeh, A.S. Shahabi, Z. Wu, Effective text classification using BERT, MTM LSTM, and DT, Data Knowl. Eng. 151 (2024) 102306.
[28] S. Zhang, N. Ran, Contrastive learning based on linguistic knowledge and adaptive augmentation for text classification, Knowl.-Based Syst. 300 (2024) 112189.
[29] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: North American Chapter of the Association for Computational Linguistics, 2019, pp. 4171–4186.
[30] H. Ye, R. Sunderraman, S. Ji, MatchXML: An efficient text-label matching framework for extreme multi-label text classification, IEEE Trans. Knowl. Data Eng. 36 (2023) 4781–4793.
[31] N. Mylonas, I. Mollas, G. Tsoumakas, An attention matrix for every decision: Faithfulness-based arbitration among multiple attention-based interpretations of transformers in text classification, Data Min. Knowl. Discov. 38 (1) (2024) 128–153.
[32] Y. Zhou, P. Xu, X. Liu, B. An, W. Ai, F. Huang, Explore spurious correlations at the concept level in language models for text classification, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 478–492.
[33] S.R. Kasa, A. Goel, K. Gupta, S. Roychowdhury, A. Bhanushali, N. Pattisapu, P.S. Murthy, Exploring ordinality in text classification: A comparative study of explicit and implicit techniques, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 5390–5404.
[34] Z. Guo, Z. Liu, Z. Ling, S. Wang, L. Jin, Y. Li, Text classification by contrastive learning and cross-lingual data augmentation for alzheimer's disease detection, in: International Conference on Computational Linguistics, 2020.
[35] M. Bayer, M.-A. Kaufhold, C. Reuter, A survey on data augmentation for text classification, ACM Comput. Surv. 55 (2021) 1–39.
[36] R. Liu, W.-J. Liang, W. Luo, Y. Song, H. Zhang, R. Xu, Y. Li, M. Liu, Recent advances in hierarchical multi-label text classification: A survey, 2023, arXiv abs/2307.16265.

[37] L.S. da Costa, I.L. Oliveira, R. Fileto, Text classification using embeddings: a survey, Knowl. Inf. Syst. 65 (2023) 2761–2803.
[38] T. Lin, Y. Wang, X. Liu, X. Qiu, A survey of transformers, AI Open 3 (2021) 111–132.
[39] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, J. rong Wen, A survey of large language models, 2023, arXiv abs/2303.18223.
[40] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, J. Li, A survey of knowledge-enhanced pre-trained language models, 2022, arXiv abs/2211.05994.
[41] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, J. Li, A survey of knowledge enhanced pre-trained language models, IEEE Trans. Knowl. Data Eng. 36 (4) (2024) 1413–1430.
[42] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, Sci. China Technol. Sci. 63 (2020) 1872–1897.
[43] C. Liu, H. Zhang, K. Zhao, X. Ju, L. Yang, LLMEmbed: Rethinking lightweight LLM's genuine function in text classification, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 7994–8004.
[44] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T.J. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020, arXiv abs/2005.14165.
[45] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, C. Xu, Transformers in computational visual media: A survey, Comput. Vis. Media 8 (2021) 33–62.
[46] J. Selva, A.S. Johansen, S. Escalera, K. Nasrollahi, T.B. Moeslund, A. Clap'es, Video transformers: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2023) 12922–12943.
[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[48] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 2227–2237.
[49] Z. Lu, J. Tian, W. Wei, X. Qu, Y. Cheng, W. Xie, D. Chen, Mitigating boundary ambiguity and inherent bias for text classification in the era of large language models, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 7841–7864.
[50] H. Rathnayake, J. Sumanapala, R. Rukshani, S. Ranathunga, AdapterFusion-based multi-task learning for code-mixed and code-switched text classification, Eng. Appl. Artif. Intell. 127 (2024) 107239.
[51] L. Yu, H. Li, K. Chen, L. Shou, BoKA: Bayesian optimization based knowledge amalgamation for multi-unknown-domain text classification, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 4035–4046.
[52] P. Vijayaraghavan, H. Wang, L. Shi, T. Baldwin, D. Beymer, E. Degan, Self-regulated data-free knowledge amalgamation for text classification, in: North American Chapter of the Association for Computational Linguistics, 2024, pp. 491–502.
[53] E. Villa-Cueva, A.P. L'opez-Monroy, F. S'anchez-Vega, T. Solorio, Adaptive cross-lingual text classification through in-context one-shot demonstrations, in: North American Chapter of the Association for Computational Linguistics, 2024, pp. 8317–8335.
[54] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: Annual Conference on Neural Information Processing Systems, 2014, pp. 2204–2212.
[55] D. Bahdanau, K.H. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015.
[56] A. Vaswani, N.M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Annual Conference on Neural Information Processing Systems, 2017, pp. 5998–6008.
[57] Z. Liang, J. Guo, W. Qiu, Z. Huang, S. Li, When graph convolution meets double attention: online privacy disclosure detection with multi-label text classification, Data Min. Knowl. Discov. 38 (3) (2024) 1171–1192.
[58] G. Wang, Y. Du, Y. Jiang, J. Liu, X. Li, X. Chen, H. Gao, C. Xie, Y.-l. Lee, Label-text bi-attention capsule networks model for multi-label text classification, Neurocomputing 588 (2024) 127671.
[59] Y. Wu, Y. Liu, Z. Zhao, W. Lu, Y. Zhang, C. Sun, F. Wu, K. Kuang, De-biased attention supervision for text classification with causality, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 19279–19287.
[60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 770–778.
[61] N. Zhou, N. Yao, N. Hu, J. Zhao, Y. Zhang, CDGAN-BERT: Adversarial constraint and diversity discriminator for semi-supervised text classification, Knowl.-Based Syst. 284 (2024) 111291.
[62] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, in: 8th International Conference on Learning Representations, 2020.
[63] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced language representation with informative entities, in: Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1441–1451.
[64] K. Clark, M.-T. Luong, Q.V. Le, C.D. Manning, ELECTRA: Pre-training text encoders as discriminators rather than generators, in: International Conference on Learning Representations, 2020.
[65] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, Adv. Neural Inf. Process. Syst. 33 (2020) 16857–16867.
[66] Z. Dai, G. Lai, Y. Yang, Q. Le, Funnel-transformer: Filtering out sequential redundancy for efficient language processing, Adv. Neural Inf. Process. Syst. 33 (2020) 4271–4282.
[67] R. Al-Rfou, D. Choe, N. Constant, M. Guo, L. Jones, Character-level language modeling with deeper self-attention, in: AAAI Conference on Artificial Intelligence, 2018.
[68] Z. Dai, Z. Yang, Y. Yang, J.G. Carbonell, Q.V. Le, R. Salakhutdinov, Transformer-XL: Attentive language models beyond a fixed-length context, in: Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2978–2988.
[69] Z. Yang, Z. Dai, Y. Yang, J.G. Carbonell, R. Salakhutdinov, Q.V. Le, XLNet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, 2019, pp. 5754–5764.
[70] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. rahman Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Annual Meeting of the Association for Computational Linguistics, 2019, pp. 7871–7880.
[71] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (140) (2020) 1–67.
[72] K. Dong, B. Jiang, H. Li, Z. Zhu, P. Liu, Meta-learning triplet contrast network for few-shot text classification, Knowl.-Based Syst. (2024) 112440.
[73] W. Liang, T. Zhang, H. Liu, F. Zhang, SELP: A semantically-driven approach for separated and accurate class prototypes in few-shot text classification, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 9732–9741.
[74] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C.L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L.E. Miller, M. Simens, A. Askell, P. Welinder, P.F. Christiano, J. Leike, R.J. Lowe, Training language models to follow instructions with human feedback, 2022, arXiv abs/2203.02155.
[75] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, D. Yang, Is ChatGPT a general-purpose natural language processing task solver? 2023, arXiv preprint arXiv:2302.06476.
[76] X. Lv, Few-shot text classification with an efficient prompt tuning method in meta-learning framework, Int. J. Pattern Recognit. Artif. Intell. 38 (03) (2024) 2451006.
[77] S. Xiong, Y. Zhao, J. Zhang, L. Mengxiang, Z. He, X. Li, S. Song, Dual prompt tuning based contrastive learning for hierarchical text classification, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 12146–12158.
[78] L. Dai, Y. Yin, E. Chen, H. Xiong, Unifying graph retrieval and prompt tuning for graph-grounded text classification, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2682–2686.
[79] J.W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, et al., Scaling language models: Methods, analysis & insights from training gopher, 2021, arXiv preprint arXiv:2112.11446.
[80] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, et al., Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model, 2022, arXiv preprint arXiv:2201.11990.
[81] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, J. Mach. Learn. Res. 24 (240) (2023) 1–113.
[82] T. Mullen, N. Collier, Sentiment analysis using support vector machines with diverse information sources, in: Conference on Empirical Methods in Natural Language Processing, 2004, pp. 412–418.
[83] S. Tan, X. Cheng, Y. Wang, H. Xu, Adapting naive Bayes to domain adaptation for sentiment analysis, in: European Conference on Information Retrieval, 2009, pp. 337–349.
[84] S.V. Wawre, S.N. Deshmukh, Sentiment classification using machine learning techniques, Int. J. Sci. Res. (IJSR) 5 (4) (2016) 819–821.
[85] A. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 142–150.
[86] B. Trstenjak, S. Mikac, D. Donko, KNN with TF-IDF based framework for text categorization, Procedia Eng. 69 (2014) 1356–1364.

[87] R. Johnson, T. Zhang, Deep pyramid convolutional neural networks for text categorization, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 562–570.
[88] L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: A survey, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8 (4) (2018) e1253.
[89] R. Behzadidoost, F. Mahan, H. Izadkhah, Granular computing-based deep learning for text classification, Inform. Sci. 652 (2024) 119746.
[90] B. Ma, E. Lai, W.Q. Yan, J. Wu, A privacy-preserving word embedding text classification model based on privacy boundary constructed by deep belief network, Multimedia Tools Appl. 83 (10) (2024) 30181–30206.
[91] W. Tan, N.D. Nguyen, L. Du, W. Buntine, Harnessing the power of beta scoring in deep active learning for multi-label text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 15240–15248.
[92] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[93] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.
[94] Z. Zheng, Y.-C. Zhou, K.-Y. Chen, X.-Z. Lu, Z.-T. She, J.-R. Lin, A text classification-based approach for evaluating and enhancing the machine interpretability of building codes, Eng. Appl. Artif. Intell. 127 (2024) 107207.
[95] Z. Yang, F. Emmert-Streib, Optimal performance of binary relevance CNN in targeted multi-label text classification, Knowl.-Based Syst. 284 (2024) 111286.
[96] P.M. Domingos, A few useful things to know about machine learning, Commun. ACM 55 (2012) 78–87.
[97] S. Kiritchenko, X.-D. Zhu, S.M. Mohammad, Sentiment analysis of short informal texts, J. Artificial Intelligence Res. 50 (2014) 723–762.
[98] L. Qu, G. Ifrim, G. Weikum, The bag-of-opinions method for review rating prediction from sparse text patterns, in: International Conference on Computational Linguistics, 2010, pp. 913–921.
[99] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.
[100] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: The 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 655–665.
[101] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.
[102] R. Johnson, T. Zhang, Effective use of word order for text categorization with convolutional neural networks, in: North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 103–112.
[103] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.
[104] D. Tang, B. Qin, T. Liu, Learning semantic representations of users and products for document level sentiment classification, in: Annual Meeting of the Association for Computational Linguistics, 2015, pp. 1014–1023.
[105] Y. Kim, Y. Jernite, D.A. Sontag, A.M. Rush, Character-aware neural language models, in: AAAI Conference on Artificial Intelligence, 2015, pp. 2741–2749.
[106] B. Liu, Y. Zhou, W.-X. Sun, Character-level text classification via convolutional neural network and gated recurrent unit, Int. J. Mach. Learn. Cybern. 11 (2020) 1939–1949.
[107] T. Londt, X. Gao, P. Andreae, Evolving character-level densenet architectures using genetic programming, in: International Conference on the Applications of Evolutionary Computation, Part of EvoStar, Springer, 2021, pp. 665–680.
[108] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2873–2879.
[109] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: AAAI Conference on Artificial Intelligence, 2015, pp. 2267–2273.
[110] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E.H. Hovy, Hierarchical attention networks for document classification, in: North American Chapter of the Association for Computational Linguistics, 2016, pp. 1480–1489.
[111] L. Qin, Z. Zhang, H. Zhao, Z. Hu, E. Xing, Adversarial connective-exploiting networks for implicit discourse relation classification, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1006–1017.
[112] P. Liu, X. Qiu, X. Huang, Adversarial multi-task learning for text classification, in: Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1–10.
[113] Y. Li, J. Ye, Learning adversarial networks for semi-supervised text classification via policy gradient, in: Proceedings of the 24th Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1715–1723.
[114] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 7370–7377.
[115] X. Liu, X. You, X. Zhang, J. Wu, P. Lv, Tensor graph convolutional networks for text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 8409–8416.
[116] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, L. Wang, Every document owns its structure: Inductive text classification via graph neural networks, in: Annual Meeting of the Association for Computational Linguistics, 2020, pp. 334–339.
[117] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in: Advances in Neural Information Processing Systems, 2017, pp. 3859–3869.
[118] Y. Wang, A. Sun, J. Han, Y. Liu, X. Zhu, Sentiment analysis by capsules, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1165–1174.
[119] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, Z. Zhao, Investigating capsule networks with dynamic routing for text classification, in: Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3110–3119.
[120] M. Yang, W. Zhao, L. tai Chen, Q. Qu, Z. Zhao, Y. Shen, Investigating the transferring capability of capsule networks for text classification, Neural Netw. 118 (2019) 247–261.
[121] W. Zhao, H. Peng, S. Eger, E. Cambria, M. Yang, Towards scalable and reliable capsule networks for challenging nlp applications, in: Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1549–1559.
[122] Z. Chen, T. Qian, Transfer capsule network for aspect level sentiment classification, in: Annual Meeting of the Association for Computational Linguistics, 2019, pp. 547–556.
[123] M. Li, J. Zhu, Y. Wang, Y. Yang, Y. Li, H. Wang, RulePrompt: Weakly supervised text classification with prompting PLMs and self-iterative logical rules, in: Proceedings of the ACM on Web Conference 2024, 2024, pp. 4272–4282.
[124] F. Deng, J. Zhong, N. Li, L. Fu, B. Jiang, Y. Ningbo, F. Qi, H. Xin, T.L. Lam, Text-guided graph temporal modeling for few-shot video classification, Eng. Appl. Artif. Intell. 137 (2024) 109076.
[125] K. Feng, L. Huang, K. Wang, W. Wei, R. Zhang, Prompt-based learning framework for zero-shot cross-lingual text classification, Eng. Appl. Artif. Intell. 133 (2024) 108481.
[126] S. Zhang, N. Ran, Fine-grained and coarse-grained contrastive learning for text classification, Neurocomputing 596 (2024) 128084.
[127] S. Roychowdhury, K. Gupta, S.R. Kasa, P. Srinivasa Murthy, Tackling concept shift in text classification using entailment-style modeling, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 5647–5656.
[128] Y. Zhang, Multi-granular text classification with minimal supervision, in: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024, pp. 1158–1160.
[129] C. Junfan, R. Zhang, Y. Zheng, Q. Chen, C. Hu, Y. Mao, DualCL: Principled supervised contrastive learning as mutual information maximization for text classification, in: Proceedings of the ACM on Web Conference 2024, 2024, pp. 4362–4371.
[130] H. Chen, G. Zheng, Y. Ji, Generating hierarchical explanations on text classification via feature interaction detection, in: Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5578–5593.
[131] D. Croce, G. Castellucci, R. Basili, GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples, in: Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2114–2119.
[132] Q. Qin, W. Hu, B. Liu, Feature projection for improved text classification, in: Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8161–8171.
[133] L. Paletto, V. Basile, R. Esposito, Label augmentation for zero-shot hierarchical text classification, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 7697–7706.
[134] X. Zhao, Y. An, N. Xu, X. Geng, Variational continuous label distribution learning for multi-label text classification, IEEE Trans. Knowl. Data Eng. 36 (2024) 2716–2729.
[135] I. Chalkidis, M. Fergadiotis, S. Kotitsas, P. Malakasiotis, N. Aletras, I. Androutsopoulos, An empirical study on large-scale multi-label text classification including few and zero-shot labels, in: Conference on Empirical Methods in Natural Language Processing, 2020, pp. 7503–7515.
[136] D. Mekala, J. Shang, Contextualized weak supervision for text classification, in: Annual Meeting of the Association for Computational Linguistics, 2020, pp. 323–333.
[137] G. Paolini, B. Athiwaratkun, J. Krone, M. Jie, A. Achille, R. Anubhai, C.N. dos Santos, B. Xiang, S. Soatto, et al., Structured prediction as translation between augmented natural languages, in: 9th International Conference on Learning Representations, 2021, pp. 1–26.
[138] S. Hu, N. Ding, H. Wang, Z. Liu, J.-Z. Li, M. Sun, Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification, in: Annual Meeting of the Association for Computational Linguistics, 2021, pp. 2225–2240.
[139] Q. Chen, R. Zhang, Y. Zheng, Y. Mao, Dual contrastive learning: Text classification via label-aware data augmentation, 2022, arXiv abs/2201.08702.
[140] A. Mueller, J. Krone, S. Romeo, S. Mansour, E. Mansimov, Y. Zhang, D. Roth, Label semantic aware pre-training for few-shot text classification, in: Annual Meeting of the Association for Computational Linguistics, 2022, pp. 8318–8334.

[141] K. Guo, H. Yu, C. Liao, J. Li, H. Zhang, ZeroAE: Pre-trained language model based autoencoder for transductive zero-shot text classification, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 3202–3219.
[142] W. Yang, R. Zhang, J. Chen, L. Wang, J. Kim, Prototype-guided pseudo labeling for semi-supervised text classification, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 16369–16382.
[143] H.S. Lee, S. Choi, Y. Lee, H. Moon, S. Oh, M. Jeong, H. Go, C. Wallraven, Cross encoding as augmentation: Towards effective educational text classification, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 2184–2195.
[144] C. Clarke, Y. Heng, Y. Kang, K. Flautner, L. Tang, J. Mars, Label agnostic pre-training for zero-shot text classification, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 1009–1021.
[145] S.K. Hong, T.Y. Jang, LEA: Meta knowledge-driven self-attentive document embedding for few-shot text classification, in: North American Chapter of the Association for Computational Linguistics, 2022, pp. 99–106.
[146] J. Wang, C. Wang, F. Luo, C. Tan, M. Qiu, F. Yang, Q. Shi, S. Huang, M. Gao, Towards unified prompt tuning for few-shot text classification, in: Conference on Empirical Methods in Natural Language Processing, 2022, pp. 524–536.
[147] Z. Zhang, K. Chen, R. Wang, M. Utiyama, E. Sumita, Z. Li, H. Zhao, Universal multimodal representation for language understanding, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2023) 9169–9185.
[148] Y. Zhang, C. Yuan, X. Wang, Z. Bai, Y. Liu, Learn to adapt for generalized zero-shot text classification, in: Annual Meeting of the Association for Computational Linguistics, 2022, pp. 517–527.
[149] A. Gera, A. Halfon, E. Shnarch, Y. Perlitz, L. Ein-Dor, N. Slonim, Zero-shot text classification with self-training, in: Conference on Empirical Methods in Natural Language Processing, 2022, pp. 1107–1119.
[150] T. Zhang, Z. Xu, T. Medini, A. Shrivastava, Structural contrastive representation learning for zero-shot multi-label text classification, in: Conference on Empirical Methods in Natural Language Processing, 2022, pp. 4937–4947.
[151] S. Nishikawa, I. Yamada, Y. Tsuruoka, I. Echizen, A multilingual bag-of-entities model for zero-shot cross-lingual text classification, in: Proceedings of the 26th Conference on Computational Natural Language Learning, CoNLL, 2022, pp. 1–12.
[152] S. Min, M. Lewis, H. Hajishirzi, L. Zettlemoyer, Noisy channel language model prompting for few-shot text classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 2022, pp. 5316–5330.
[153] J. Zha, Z. Li, Y. Wei, Y. Zhang, Disentangling task relations for few-shot text classification via self-supervised hierarchical task clustering, in: Conference on Empirical Methods in Natural Language Processing, 2022, pp. 5236–5247.
[154] L. Zhao, C. Yao, EICO: Improving few-shot text classification via explicit and implicit consistency regularization, in: Annual Meeting of the Association for Computational Linguistics, 2022, pp. 3582–3587.
[155] H. Zhang, X. Zhang, H. Huang, L. Yu, Prompt-based meta-learning for few-shot text classification, in: Conference on Empirical Methods in Natural Language Processing, 2022, pp. 1342–1357.
[156] E. Shnarch, A. Gera, A. Halfon, L. Dankin, L. Choshen, R. Aharonov, N. Slonim, Cluster & tune: Boost cold start performance in text classification, in: Annual Meeting of the Association for Computational Linguistics, 2022, pp. 7639–7653.
[157] Y. Zhao, Z. Tian, H. Yao, Y. Zheng, D. Lee, Y. Song, J. Sun, N.L. Zhang, Improving meta-learning for low-resource text classification and generation via memory imitation, in: Annual Meeting of the Association for Computational Linguistics, 2022, pp. 583–595.
[158] H. Choi, D. Choi, H. Lee, Early stopping based on unlabeled samples in text classification, in: Annual Meeting of the Association for Computational Linguistics, 2022, pp. 708–718.
[159] Y. Wang, W. Wang, Q. Chen, K. Huang, A. Nguyen, S. De, Prompt-based zero-shot text classification with conceptual knowledge, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 30–38.
[160] L. Qin, W. Wang, Q. Chen, W. Che, CLIPText: A new paradigm for zero-shot text classification, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 1077–1088.
[161] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.
[162] Y.-S. Wang, T.-C. Chi, R. Zhang, Y. Yang, PESCO: Prompt-enhanced self contrastive learning for zero-shot text classification, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 14897–14911.
[163] C. Liu, W. Zhang, G. Chen, X. Wu, A.T. Luu, C.-H. Chang, L. Bing, Zero-shot text classification via self-supervised tuning, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 1743–1761.
[164] Y. Yu, Y. Zhuang, R. Zhang, Y. Meng, J. Shen, C. Zhang, ReGen: Zero-shot text classification via training data generation with progressive dense retrieval, in: Annual Meeting of the Association for Computational Linguistics, 2023, pp. 11782–11805.
[165] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168–177.
[166] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, DBpedia - A large-scale, multilingual knowledge base extracted from wikipedia, Semant. Web 6 (2015) 167–195.
[167] Q. Diao, M. Qiu, C.-Y. Wu, A. Smola, J. Jiang, C. Wang, Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS), in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 193–202.
[168] J. Wiebe, T. Wilson, C. Cardie, Annotating expressions of opinions and emotions in language, Lang. Resour. Eval. 39 (2005) 165–210.
[169] B. Pang, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, in: Annual Meeting of the Association for Computational Linguistics, 2005, pp. 115–124.
[170] B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in: Annual Meeting of the Association for Computational Linguistics, 2004, pp. 271–278.
[171] X. Li, D. Roth, Learning question classifiers, in: International Conference on Computational Linguistics, 2002, pp. 1–7.
[172] J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM conference on Recommender systems, 2013, pp. 165–172.
[173] P. He, X. Liu, J. Gao, W. Chen, Deberta: decoding-enhanced bert with disentangled attention, in: International Conference on Learning Representations, 2020.
[174] I. Beltagy, M.E. Peters, A. Cohan, Longformer: The long-document transformer, 2020, arXiv preprint arXiv:2004.05150.
[175] A. Aghajanyan, A. Gupta, A. Shrivastava, X. Chen, L. Zettlemoyer, S. Gupta, Muppet: Massive multi-task representations with pre-finetuning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 5799–5811.
[176] P. Delobelle, T. Winters, B. Berendt, RobBERT: a dutch roberta-based language model, in: Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 3255–3265.

**Yujia Wu** received the B.S. degree in Electronic Information Engineering from Hubei University of Economics in 2011, Wuhan, China. He received M.S degree in Electronic and Communication Engineering from South-central University for Nationalities in 2014, Wuhan, China. He received Ph.D. degree in Computer Science from Wuhan University in 2020, Wuhan, China. He is currently an Assistant Professor in School of Information Science and Technology, Sanda University, Shanghai, China. His main research interests include data mining and natural language processing. His works have been published in artificial intelligence journals and conferences, including Pattern Recognition, KAIS, Neurocomputing, Chinese Journal of Electronics, PRCV, ICIC, ADMA, IJCNN and so on. (wuyujia@whu.edu.cn).

**Jun Wan** received the B.S. degree from Electronic Information Engineering College, Anhui University of Finance and Economics, China, in 2010, and the Ph.D. degree in School of Computer Science, Wuhan University, China, in 2019. From 2019 to 2021, He was a Post-Doctoral Fellow with the College of Computer Science and Software Engineering, Shenzhen University, China. He is now an Associate Professor in the School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, 430073, China. His main research interests include computer vision, landmark detection and image/video captioning. His works have been published in premier computer vision journals and conferences, including IJCAI, TIP, TCYB, TKDE, TNNLS, TFS, Neural Networks, Pattern Recognition, Information Sciences and so on.