A Text Classification Survey: From Shallow to Deep Learning

8 authors, including:

Qian Li
Beihang University (BUAA)
1 PUBLICATION   **0** CITATIONS

SEE PROFILE

Hao Peng
Beihang University (BUAA)
93 PUBLICATIONS   **756** CITATIONS

SEE PROFILE

Jianxin Li
Beihang University (BUAA)
92 PUBLICATIONS   **697** CITATIONS

SEE PROFILE

Congying Xia
University of Illinois at Chicago
30 PUBLICATIONS   **274** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   A Lightweight Music Texture Transfer System View project

Project   Subtyping Parkinson's Disease with Multi-Modal Data View project

A Survey on Text Classification: From Shallow to Deep Learning

QIAN LI, Beihang University, China
HAO PENG, Beihang University, China
JIANXIN LI*, Beihang University, China
CONGYIN XIA, University of Illinois at Chicago, USA
RENYU YANG, University of Leeds, UK
LICHAO SUN, Lehigh University, USA
PHILIP S. YU, University of Illinois at Chicago, USA
LIFANG HE, Lehigh University, USA

Text classification is the most fundamental and essential task in natural language processing. The last decade has seen a surge of research in this area due to the unprecedented success of deep learning. Numerous methods, datasets, and evaluation metrics have been proposed in the literature, raising the need for a comprehensive and updated survey. This paper fills the gap by reviewing the state of the art approaches from 1961 to 2020, focusing on models from shallow to deep learning. We create a taxonomy for text classification according to the text involved and the models used for feature extraction and classification. We then discuss each of these categories in detail, dealing with both the technical developments and benchmark datasets that support tests of predictions. A comprehensive comparison between different techniques, as well as identifying the pros and cons of various evaluation metrics are also provided in this survey. Finally, we conclude by summarizing key implications, future research directions, and the challenges facing the research area.

1 INTRODUCTION

Text classification – the procedure of designating pre-defined labels for text – is an essential and significant task in many Natural Language Processing (NLP) applications, such as sentiment analysis [1][2] [3], topic labeling [4] [5] [6], question answering [7] [8] and dialog act classification [9]. In

---

*Corresponding author

Authors' addresses: Qian Li, Beihang University, Haidian, Beijing, China, liqian@act.buaa.edu.cn; Hao Peng, Beihang University, Haidian, Beijing, China, penghao@act.buaa.edu.cn; Jianxin Li, Beihang University, Haidian, Beijing, China, lijx@act.buaa.edu.cn; Congyin Xia, University of Illinois at Chicago, Chicago, IL, USA, cxia8@uic.edu; Renyu Yang, University of Leeds, Leeds, England, UK, r.yang1@leeds.ac.uk; Lichao Sun, Lehigh University, Bethlehem, PA, USA, james.lichao.sun@gmail.com; Philip S. Yu, University of Illinois at Chicago, Chicago, IL, USA, psyu@uic.edu; Lifang He, Lehigh University, Bethlehem, PA, USA, lih319@lehigh.edu.
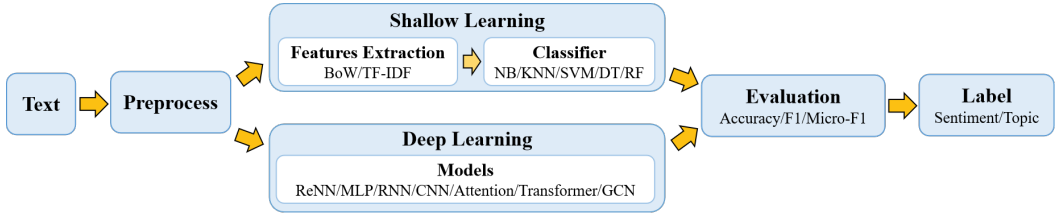

Fig. 1. Flowchart of the text classification with classic methods in each module. It is crucial to extract essential features for shallow models, but features can be extracted automatically by DNNs.

the era of information explosion, it is time-consuming and challenging to process and classify large amounts of text data manually. Besides, the accuracy of manual text classification can be easily influenced by human factors, such as fatigue and expertise. It is desirable to use machine learning methods to automate the text classification procedure to yield more reliable and less subjective results. Moreover, this can also help enhance information retrieval efficiency and alleviate the problem of information overload by locating the required information.

Fig. 1 illustrates a flowchart of the procedures involved in the text classification, under the light of shallow and deep analysis. Text data is different from numerical, image, or signal data. It requires NLP techniques to be processed carefully. The first important step is to preprocess text data for the model. Shallow learning models usually need to obtain good sample features by artificial methods and then classify them with classic machine learning algorithms. Therefore, the effectiveness of the method is largely restricted by feature extraction. However, different from shallow models, deep learning integrates feature engineering into the model fitting process by learning a set of nonlinear transformations that serve to map features directly to outputs.

The schematic illustration of the primary text classification methods is shown in Fig. 2. From the 1960s until the 2010s, shallow learning-based text classification models dominated. Shallow learning means statistics-based models, such as NaÃŕve Bayes (NB) [10], K-nearest neighbor (KNN) [11], and support vector machine (SVM) [12]. Comparing with the earlier rule-based methods, this method has obvious advantages in accuracy and stability. However, these approaches still need to do feature engineering, which is time-consuming and costly. Besides, they usually disregard the natural sequential structure or contextual information in textual data, making it challenging to learn the semantic information of the words. Since the 2010s, text classification has gradually changed from shallow learning models to deep learning models. Compared with the methods based on shallow learning, deep learning methods avoid designing rules and features by humans and automatically provide semantically meaningful representations for text mining. Therefore, most of the text classification research works are based on DNNs, which are data-driven approaches with high computational complexity. Few works focus on shallow learning models to settle the limitations of computation and data.

1.1 Major Differences and Contributions

In the literature, Kowsari et al. [13] surveyed different text feature extraction, dimensionality reduction methods, basic model structure for text classification, and evaluation methods. Minaee et al. [14] reviewed recent deep learning based text classification methods, benchmark datasets, and evaluation metrics. Unlike existing text classification surveys, we conclude existing models from shallow to deep learning with works of recent years. Shallow learning models emphasize the feature extraction and classifier design. Once the text has well-designed characteristics, it can be quickly converged by training the classifier. DNNs can perform feature extraction automatically
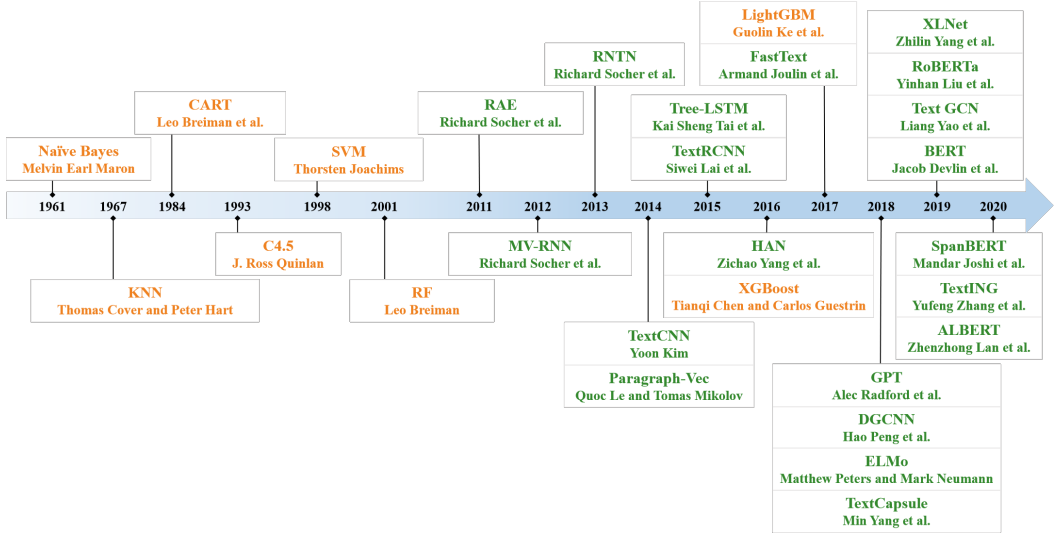
Fig. 2. Schematic illustration of the primary text classification methods from 1961 to 2020. Before 2010, almost all existing methods are based on shallow models (orange color); since 2010, most work in this area has concentrated on deep learning schemes (green color).

and learn well without domain knowledge. We then give the datasets and evaluation metrics for single-label and multi-label tasks and summarize future research challenges from data, models, and performance perspective. Moreover, we summarize various information in four tables, including the necessary information of classic shallow and deep learning models, technical details of DNNs, primary information of main datasets, and a general benchmark of state-of-the-art methods under different applications. In summary, this study's main contributions are as follows:

- We introduce the process and development of text classification and summarize the necessary information of classic models in terms of publishing years in Table 1, including venues, applications, citations, and code links.
- We present comprehensive analysis and research on primary models – from shallow to deep learning models – according to their model structures. We summarize classic or more specific models and primarily outline the design difference in terms of basic models, metric and experimental datasets in Table 2.
- We introduce the present datasets and give the formulation of main evaluation metrics, including single-label and multi-label text classification tasks. We summarize the necessary information of primary datasets in Table 3, including the number of categories, average sentence length, the size of each dataset, related papers and data addresses.
- We summarize classification accuracy scores of classical models on benchmark datasets in Table 5 and conclude the survey by discussing the main challenges facing the text classification and key implications stemming from this study.

1.2 Organization of the Survey

The rest of the survey is organized as follows. Section 2 summarizes the existing models related to text classification, including shallow learning and deep learning models. Section 3 introduces the primary datasets with a summary table and evaluation metrics on single-label and multi-label tasks. We then give quantitative results of the leading models in classic text classification datasets

in Section 4. Finally, we summarize the main challenges for deep learning text classification in Section 5 before concluding the article in Section 6.

2 TEXT CLASSIFICATION METHODS

Text classification is referred to as extracting features from raw text data and predicting the categories of text data based on such features. Numerous models have been proposed in the past few decades for text classification, as shown in Table 1. We tabulate primary information – including venues, applications, citations, and code links – of main models for text classification. The applications in this table include sentiment analysis (SA), topic labeling (TL), news classification (NC), question answering (QA), dialog act classification (DAC), natural language inference (NLI), and event prediction (EP). For shallow learning models, NB [10] is the first model used for the text classification task. Whereafter, generic classification models are proposed, such as KNN, SVM [12], and RF [15], which are called classifiers, widely used for text classification. Recently, XGBoost [16] and LightGBM [17] have arguably the potential to provide excellent performance. For deep learning models, TextCNN [18] has the highest number of references in these models, wherein a CNN model has been introduced to solve the text classification problem for the first time. While not specifically designed for handling text classification tasks, BERT [19] has been widely employed when designing text classification models, considering its effectiveness on numerous text classification datasets.

2.1 Shallow Learning Models

Shallow learning models accelerate text classification with improved accuracy and make the application scope of shallow learning expand. The first thing is to preprocess the raw input text for training shallow learning models, which generally consists of word segmentation, data cleaning, and data statistics. Then, text representation aims to express preprocessed text in a form that is much easier for computers and minimizes information loss, such as Bag-of-words (BOW), N-gram, term frequency-inverse document frequency (TF-IDF) [93], word2vec [94] and GloVe [95]. At the core of the BOW is representing each text with a dictionary-sized vector. The individual value of the vector denotes the word frequency corresponding to its inherent position in the text. Compared to BOW, N-gram considers the information of adjacent words and builds a dictionary by considering the adjacent words. TF-IDF [93] uses the word frequency and inverses the document frequency to model the text. The word2vec [94] employs local context information to obtain word vectors. The GloVe [95] – with both the local context and global statistical features – trains on the nonzero elements in a word-word co-occurrence matrix. Finally, the represented text is fed into the classifier according to selected features. Here, we discuss some of the representative classifiers in detail:

2.1.1 PGM-based methods. Probabilistic graphical models (PGMs) express the conditional dependencies among features in graphs, such as the Bayesian network [96], the hidden Markov network [97]. Such models are combinations of probability theory and graph theory.

NaÃŕve Bayes (NB) [10] is the simplest and most broadly used model based on applying Bayes' theorem. The NB algorithm has an independent assumption: when the target value has been given, the conditions between features $x = [x_1, x_2, \cdots, x_n]$ are independent (see Fig. 3). The NB algorithm primarily uses the prior probability to calculate the posterior probability. Due to its simple structure, NB is broadly used for text classification tasks. Although the assumption that the features are independent is sometimes not actual, it substantially simplifies the calculation process and performs better. To improve the performance on smaller categories, Schneider [98] proposes a feature selection score method through calculating KL-divergence [99] between the training set and corresponding categories for multinomial NB text classification. Dai et al. [100] propose a transfer learning method named Naive Bayes Transfer Classification (NBTC) to settle

Table 1. Necessary information of the principal models for text classification and the citation is counted on June 8, 2020.

Year	Method	Venue	Applications	Citations	Code Link
1961	NB [10]	JACM	TL	612	[20]
1967	KNN [11]	IEEE Trans.	-	12152	[21]
1984	CART [22]	Wadsworth	-	45967	[23]
1993	C4.5 [24]	Morgan Kaufmann	-	37847	[25]
1995	AdaBoost [26]	EuroCOLT	-	19372	[27]
1998	SVM [12]	ECML	-	10770	[20]
2001	RF [15]	Mach. Learn.	-	60249	[28]
2011	RAE [29]	EMNLP	SA, QA	1231	[30]
2012	MV-RNN [31]	EMNLP	SA	1141	[32]
2013	RNTN [33]	EMNLP	SA	3725	[34]
2014	Paragraph-Vec [35]	ICML	SA, QA	5679	[36]
2014	DCNN [7]	ACL	SA, QA	2433	[37]
2014	TextCNN [18]	EMNLP	SA, QA	7171	[38]
2015	TextRCNN [39]	AAAI	SA, TL	1141	[40]
2015	DAN [41]	ACL	SA, QA	467	[42]
2015	Tree-LSTM [2]	ACL	SA	1761	[43]
2015	CharCNN [5]	NeurIPS	SA, QA, TL	2114	[44]
2016	XGBoost [16]	KDD	QA	6187	[45]
2016	HAN [46]	NAACL	SA, TL	1889	[47]
2016	Multi-Task [48]	IJCAI	SA	410	[49]
2016	LSTMN [50]	EMNLP	SA	449	[51]
2017	LightGBM [17]	NeurIPS	QA	1065	[52]
2017	FastText [53]	EACL	SA, TL	1954	[54]
2017	Miyato et al. [55]	ICLR	SA	246	[56]
2017	TopicRNN [57]	ICML	SA	113	[58]
2017	DPCNN [59]	ACL	SA, TL	156	[60]
2017	IAN [61]	IJCAI	SA	222	[62]
2017	DeepMoji [63]	EMNLP	SA	260	[64]
2017	RAM [65]	EMNLP	SA	225	[62]
2018	ELMo [66]	NAACL	SA, QA, NLI	3722	[67]
2018	DGCNN [68]	TheWebConf	TL	81	[69]
2018	ULMFiT [70]	ACL	SA, TL, News	819	[71]
2018	LEAM [72]	ACL	TL, News	87	[73]
2018	SGM [74]	COLING	TL	42	[75]
2018	SGNN [76]	IJCAI	EP	26	[77]
2018	TextCapsule [78]	EMNLP	SA, QA, TL	118	[79]
2018	MGAN [80]	EMNLP	SA	46	[62]
2019	TextGCN [6]	AAAI	SA, TL	114	[81]
2019	BERT [19]	NAACL	SA, QA	5532	[82]
2019	MT-DNN [83]	ACL	SA, NLI	186	[84]
2019	XLNet [85]	NeurIPS	SA, QA, NC	652	[86]
2019	RoBERTa [87]	arXiv	SA, QA	203	[88]
2020	ALBERT[89]	ICLR	SA, QA	197	[90]
2020	SpanBERT [91]	TACL	QA	63	[92]

the different distribution between the training set and the target set. It uses the EM algorithm [101] to obtain a locally optimal posterior hypothesis on the target set.
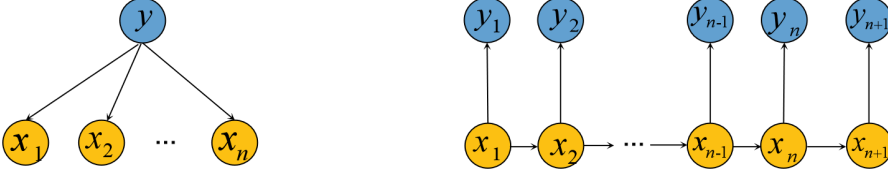

Fig. 3. The structure of NB (left) and the structure of HMM (right).

Hidden Markov model (HMM) is a Markov model assumed to be a Markov process within hidden states [97]. It is suitable for sequential text data, effective in reducing algorithmic complexity by redesigning model structure. HMM operates under the assumption that a separate process Y exists, and its behavior depends upon X. The reachable learning goal is to learn about X by observing Y, considering the state dependencies (see Fig. 3). To consider the contextual information among pages in a text, Frasconi et al. [102] reshape a text into sequences of pages and exploit the serial order relationship among pages within a text for multi-page texts. However, these methods get no excellent performance for domain text. Motivated by this, Yi et al. [103] use prior knowledge – primarily stemming from a specialized subject vocabulary set Medical Subject Headings (MeSH) [104] – to carry out the medical text classification task.
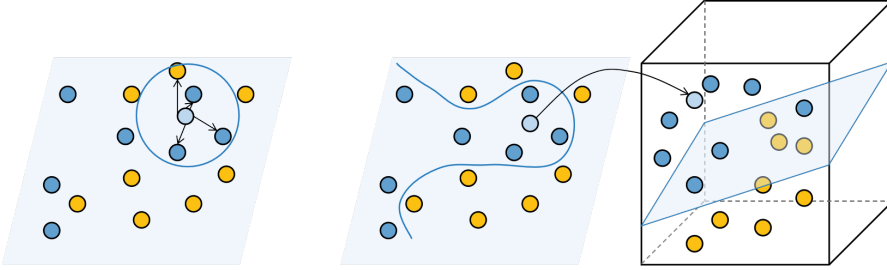

Fig. 4. The structure of KNN where $k = 4$ (left) and the structure of SVM (right). Different colored nodes represent different categories.

2.1.2   KNN-based Methods. At the core of the K-Nearest Neighbors (KNN) algorithm [11] is to classify an unlabeled sample by finding the category with most samples on the k-nearest labeled samples. It is a simple classifier without building the model and can decrease complexity through the fasting process of getting k nearest neighbors. Fig. 4 showcases the structure of KNN. We can find k training texts approaching a specific text to be classified through estimating the in-between distance. Hence, the text can be divided into the most common categories found in k training set texts. However, due to the positive correlation between model time/space complexity and the amount of data, the KNN algorithm takes an unusually long time on the large-scale datasets. To decrease the number of selected features, Soucy et al. [105] propose a KNN algorithm without feature weighting. It manages to find relevant features, building the inter-dependencies of words by using a feature selection. When the data is extremely unevenly distributed, KNN tends to classify samples with more data. The neighbor-weighted K-nearest neighbor (NWKNN) [106] is proposed to improve classification performance on the unbalanced corpora. It casts a significant weight for neighbors in a small category and a small weight for neighbors in a broad class.

2.1.3 SVM-based Methods. Cortes and Vapnik propose Support Vector Machine (SVM) [107] to tackle the binary classification of pattern recognition. Joachims [12], for the first time, uses the SVM method for text classification representing each text as a vector. As illustrated in Fig. 4, SVM-based approaches turn text classification tasks into multiple binary classification tasks. In this context, SVM constructs an optimal hyperplane in the one-dimensional input space or feature space, maximizing the distance between the hyperplane and the two categories of training sets, thereby achieving the best generalization ability. The goal is to make the distance of the category boundary along the direction perpendicular to the hyperplane is the largest. Equivalently, this will result in the lowest error rate of classification. Constructing an optimal hyperplane can be transformed into a quadratic programming problem to obtain a globally optimal solution. Choosing the appropriate kernel function is of the utmost importance to ensure SVM can deal with nonlinear problems and become a robust nonlinear classifier. To analyze what the SVM algorithms learn and what tasks are suitable, Joachims [108] proposes a theoretical learning model combining the statistical traits with the generalization performance of an SVM analyzing the features and benefits using a quantitative approach. Transductive Support Vector Machine (TSVM) [109] is proposed to lessen misclassifications of the particular test collections with a general decision function considering a specific test set. It uses prior knowledge to establish a more suitable structure and study faster.
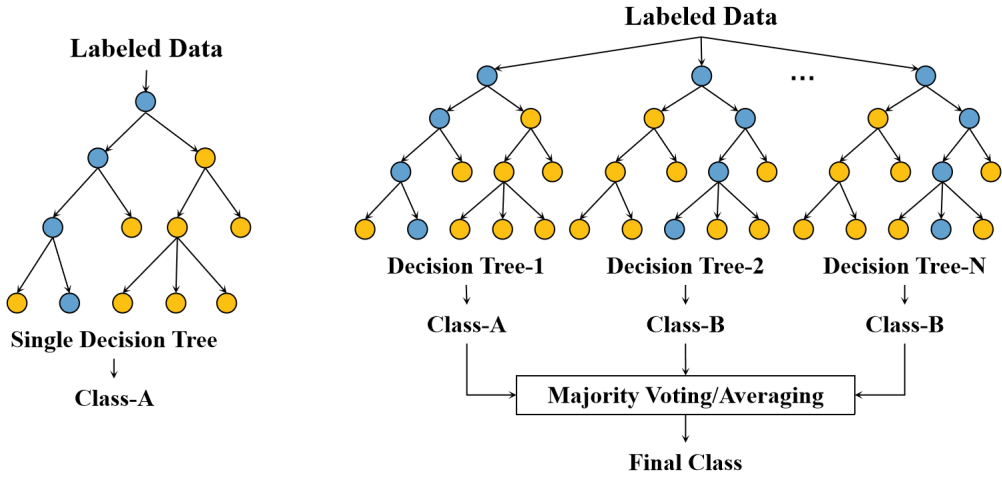
Fig. 5. The structure of DT (left) and the structure of RF (right). Nodes in blue represent the nodes of the decision route.

2.1.4 DT-based Methods. Decision Trees (DT) [110] is a supervised tree structure learning method – reflective of the idea of divide-and-conquer – and is constructed recursively. It learns disjunctive expressions and has robustness for the text with noise. As shown in Fig. 5, decision trees can be generally divided into two distinct stages: tree construction and tree pruning. It starts at the root node and tests the data samples (composed of instance sets, which have several attributes), and divides the dataset into diverse subsets according to different results. A subset of datasets constitutes a child node, and every leaf node in the decision tree represents a category. Constructing the decision tree is to determine the correlation between classes and attributes, further exploited to predict the record categories of unknown forthcoming types. The classification rules generated by the decision tree algorithm are straight-forward, and the pruning strategy can also help reduce the influence of noise. Its limitation, however, mainly derives from inefficiency in coping with

explosively increasing data size. More specifically, the ID3 [111] algorithm uses information gain as the attribute selection criterion in the selection of each node – It is used to select the attribute of each branch node, and then select the attribute having the maximum information gain value to become the discriminant attribute of the current node. Based on ID3, C4.5 [24] learns to obtain a map from attributes to classes, which effectively classifies entities unknown to new categories. DT based algorithms usually need to train for each dataset, which is low efficiency. Thus, Johnson et al. [112] propose a DT-based symbolic rule system. The method represents each text as a vector calculated by the frequency of each word in the text and induces rules from the training data. The learning rules are used for classifying the other data being similar to the training data. Furthermore, to reduce the computational costs of DT algorithms, fast decision-tree (FDT) [113] uses two-pronged strategy: pre-selecting a feature set and training multiple DTs on different data subsets. Results from multiple DTs are combined through a data-fusion technique to resolve the cases of imbalanced classes.

2.1.5 Integration-based Methods. Integrated algorithms aim to aggregate the results of multiple algorithms for better performance and interpretation. Conventional integrated algorithms are bootstrap aggregation, such as random forest (RF) [15], boosting such as AdaBoost [26], and XGBoost [16] and stacking. The bootstrap aggregation method trains multiple classifiers without strong dependencies and then aggregates their results. For instance, RF [15] consists of multiple tree classifiers wherein all trees depend on the value of the random vector sampled independently (depicted in Fig. 5). It is worth noting that each tree within the RF shares the same distribution. The generalization error of an RF relies on the strength of each tree and the relationship among trees and will converge to a limit with the increment of tree number in the forest. In boosting based algorithms, all labeled data are trained with the same weight to initially obtain a weaker classifier. The weights of the data will then be adjusted according to the former result of the classifier. The training procedure will continue by repeating such steps until the termination condition is reached. Unlike bootstrap and boosting algorithms, stacking based algorithms break down the data into n parts and use n classifiers to calculate the input data in a cascade manner – Result from upstream classifier will feed into the downstream classifier as input. The training will terminate once a pre-defined iteration number is targeted. The integrated method can capture more features from multiple trees. However, it helps little for short text. Motivated by this, Bouaziz et al. [114] combine data enrichment – with semantics in RFs for short text classification – to overcome the deficiency of sparseness and insufficiency of contextual information. In integrated algorithms, not all classifiers learn well. It is necessary to give different weights for each classifier. To differentiate contributions of trees in a forest, Islam et al. [115] exploit Semantics Aware Random Forest (SARF) classifier, choosing features similar to the features of the same class, for extracting features and producing the prediction values.

Summary. The shallow learning method is a type of machine learning. It learns from data, which are pre-defined features that are important to the performance of prediction values. However, feature engineering is tough work. Before training the classifier, we need to collect knowledge or experience to extract features from the original text. The shallow learning methods train the initial classifier based on various textual features extracted from the raw text. Toward small datasets, shallow learning models usually present better performance than deep learning models under the limitation of computational complexity. Therefore, some researchers have studied the design of shallow models for specific domains with fewer data.

2.2 Deep Learning Models

The DNNs consist of artificial neural networks that simulate the human brain to automatically learn high-level features from data, getting better results than shallow learning models in speech recognition, image processing, and text understanding. Input datasets should be analyzed to classify the data, such as a single-label, multi-label, unsupervised, unbalanced dataset. According to the trait of the dataset, the input word vectors are sent into the DNN for training until the termination condition is reached. The performance of the training model is verified by the downstream task, such as sentiment classification, question answering, and event prediction. We show some DNNs over the years in Table 2, including designs that are different from the corresponding basic models, evaluation metrics, and experimental datasets.

As shown in Table 2, the feed-forward neural network and the recursive neural network are the first two deep learning approaches used for the text classification task, which improve performance compared with shallow learning models. Then, CNNs, RNNs, and attention mechanisms are used for text classification. Many researchers advance text classification performance for different tasks by improving CNN, RNN, and attention, or model fusion and multi-task methods. The appearance of Bidirectional Encoder Representations from Transformers (BERT) [19], which can generate contextualized word vectors, is a significant turning point in the development of text classification and other NLP technologies. Many researchers have studied text classification models based on BERT, which achieves better performance than the above models in multiple NLP tasks, including text classification. Besides, some researchers study text classification technology based on GNN [6] to capture structural information in the text, which cannot be replaced by other methods. Here, we classify DNNs by structure and discuss some of the representative models in detail:

2.2.1 ReNN-based Methods. Shallow learning models cost lots of time on design features for each task. The recursive neural network (ReNN) can automatically learn the semantics of text recursively and the syntax tree structure without feature design, as shown in Fig. 6. We give an example of ReNN based models. First, each word of input text is taken as the leaf node of the model structure. Then all nodes are combined into parent nodes using a weight matrix. The weight matrix is shared across the whole model. Each parent node has the same dimension with all leaf nodes. Finally, all nodes are recursively aggregated into a parent node to represent the input text to predict the label.
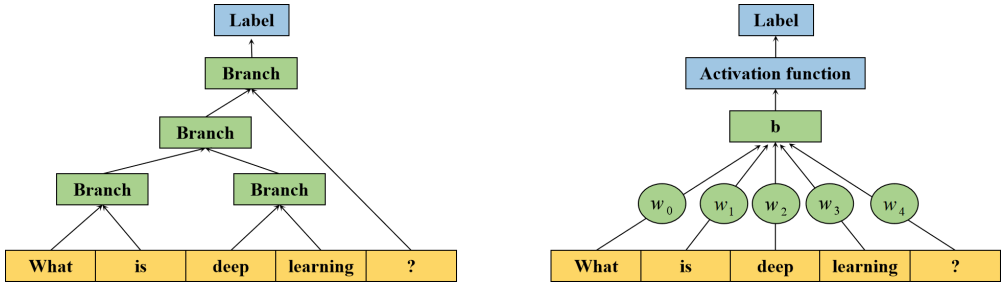
Fig. 6. The architecture of ReNN (left) and the architecture of MLP (right).

ReNN-based models improve performance compared with shallow learning models and save on labor costs due to excluding feature designs used for different text classification tasks. The recursive autoencoder (RAE) [29] is used to predict the distribution of sentiment labels for each input sentence and learn the representations of multi-word phrases. To learn compositional vector representations for each input text, the matrix-vector recursive neural network (MV-RNN) [31]

Table 2. Basic information based on different models. Trans: Transformer. Time: training time.

Model	Design	Metrics	Datasets
ReNN	recursive autoencoders [29]	Accuracy	MPQA, MR, EP
	recursive neural network [31]	Accuracy, F1	MR
	richer supervised training [33]	Accuracy	Sentiment Treebank
	multiple recursive layers [116]	Accuracy	SST-1;SST-2
MLP	a deep unordered model [41]	Accuracy, Time	RT, SST, IMDB
	paragraph vector [35]	Error Rate	SST, IMDB
RNN	tree-structured topologies [2]	Accuracy	SST-1, SST-2
	a memory cell [3]	Accuracy	SST
	RCNN and a max-pooling layer [39]	Accuracy, $Macro-F1$	20NG, Fudan, ACL, SST-2
	multi-timescale [8]	Accuracy	SST-1, SST-2, QC, IMDB
	embeddings of text regions [117]	Error Rate	IMDB, Elec, RCV1, 20NG
	2DCNN [118]	Accuracy	SST-1, SST-2, Subj, TREC, etc.
	multi-task [48]	Accuracy	SST-1, SST-2, Subj, IMDB
	distant supervision [63]	Accuracy	SS-Twitter, SE1604, etc.
	global dependencies [57]	Error Rate	IMDB
	virtual adversarial training [55]	Error Rate	IMDB, DBpedia, RCV1, etc.
	capsule [119]	Accuracy	MR, SST-1, Hospital Feedback
CNN	basic CNN [18]	Accuracy	MR, SST-1, SST-2, Subj, etc.
	dynamic k-Max pooling [7]	Accuracy	MR, TREC, Twitter
	character-level [5]	Error Rate	AG, Yelp P, DBPedia, etc.
	preceding short texts [9]	Accuracy	DSTC 4, MRDA, SwDA
	extreme multi-label [120]	P@k, DCG@k, etc.	EUR-Lex, Wiki-30K, etc.
	deep pyramid CNN [59]	Error Rate	AG, DBPedia, Yelp.P, etc.
	knowledge base [121]	Accuracy	TREC, Twitter, AG, Bing, MR
	8âĂŘbit character encoding [122]	Accuracy	Geonames toponyms, etc.
	dynamic routing [78]	Accuracy	Subj, TREC, Reuters, etc.
	hierarchical relations [123]	$Micro-F1$, $Macro-F1$, etc.	RCV1, Amazon670K
	meta-learning [124]	Accuracy	20NG, RCV, Reuters-2157, etc.
Attention	hierarchical attention [46]	Accuracy	Yelp.F, IMDB, YahooA, Amz.F
	add bilingual BiLSTM [125]	Accuracy	NLP&CC 2013 [126]
	intra-attention mechanism [50]	Accuracy	SST-1
	two-way attention mechanism [127]	P, MAP, MRR	TREC-QA, WikiQA, etc.
	Inner-Attention mechanism [128]	Accuracy	SNLI
	cross-attention mechanism [129]	F1	WebQuestion
	self-attention sentence embedding [130]	Accuracy	Yelp, Age
	sequence generation model [74]	HL, $Micro-F1$	RCV1-V2, AAPD
	deep contextualized representation [66]	Accuracy, F1	SQuAD, SNLI, SRL, SST-5, etc.
	a label tree-based model [131]	P@k, N@k, PSP@k	EUR-Lex, Amazon-670K, etc.
	knowledge powered attention [132]	Accuracy	Weibo, Product Review, etc.
	bi-directional block self-attention [133]	Accuracy, Time	CR, MPQA, SST-1, SUBJ, etc.
	deep contextualized representation [66]	Accuracy	SQuAD, SNLI, SST-5
Trans	bidirectional encoder [19]	Accuracy	SST-2, QQP, QNLI, CoLA
	multi-label legal text [134]	P@K, RP@K, R@K, etc.	EUR-LEX
	fine-tune BERT [135]	Error Rate	IMDB, TREC, DBPedia, etc.
	autoregressive pretraining [85]	DNCG@K, EM, F1, etc.	IMDB, Yelp-2, AG, MNLI, etc.
	modifications on BERT [87]	SQuAD, MNLI-m, SST-2	F1, Accuracy
	improvement of BERT [89]	F1, Accuracy	SST, MNLI, SQuAD
GNN	graph-CNN for multi-label text [68]	$Micro-F1$, $Macro-F1$, etc.	RCV1, NYTimes
	build a heterogeneous graph [6]	Accuracy	20NG, Ohsumed, R52, R8, MR
	removing the nonlinearities [136]	Accuracy, Time	20NG, R8, R52, Ohsumed, MR
	a text level graph [137]	Accuracy	R8, R52, Ohsumed
	hierarchical taxonomy-aware [138]	$Micro-F1$, $Macro-F1$	RCV1, EUR-Lex, etc.
	graph attention network-based [139]	$Micro-F1$, HL	Reuters-21578, RCV1-V2, etc.

introduces a ReNN model to learn the representation of phrases and sentences. It allows that the length and type of input texts are inconsistent. MV-RNN allocates a matrix and a vector for each node on the constructed parse tree. Furthermore, the recursive neural tensor network (RNTN) [33] is proposed with a tree structure to capture the semantics of sentences. It inputs phrases with different length and represents the phrases by parse trees and word vectors. The vectors of higher nodes on the parse tree are estimated by the equal tensor-based composition function. For RNTN, the time complexity of building the textual tree is high, and expressing the relationship between documents is complicated within a tree structure. The performance is usually improved, with the depth being increased for DNNs. Therefore, Irsoy et al. [116] propose a deep recursive neural network (DeepReNN), which stacks multiple recursive layers. It is built by binary parse trees and learns distinct perspectives of compositionality in language.

2.2.2 MLP-based Methods.
A multilayer perceptron (MLP) [140], sometimes colloquially called "vanilla" neural network, is a simple neural network structure that is used for capturing features automatically. As shown in Fig. 6, we show a three-layer MLP model. It contains an input layer, a hidden layer with an activation function in all nodes, and an output layer. Each node connects with a certain weight w_i. It treats each input text as a bag of words and achieves high performance on many text classification benchmarks comparing with shallow learning models.

There are some MLP-based methods proposed by some research groups for text classification tasks. The Paragraph Vector (Paragraph-Vec) [35] is the most popular and widely used method, which is similar to the Continuous Bag of Words (CBOW) [94]. It gets fixed-length feature representations of texts with various input lengths by employing unsupervised algorithms. Comparing with CBOW, it adds a paragraph token mapped to the paragraph vector by a matrix. The model predicts the fourth word by the connection or average of this vector to the three contexts of the word. Paragraph vectors can be used as a memory for paragraph themes and are used as a paragraph function and inserted into the prediction classifier.

2.2.3 RNN-based Methods.
The recurrent neural network (RNN) is broadly used due to capturing long-range dependency through recurrent computation. The RNN language model learns historical information, considering the location information among all words suitable for text classification tasks. We show an RNN model for text classification with a simple sample, as shown in Fig. 7. Firstly, each input word is represented by a specific vector using a word embedding technology. Then, the embedding word vectors are fed into RNN cells one by one. The output of RNN cells are the same dimension with the input vector and are fed into the next hidden layer. The RNN shares parameters across different parts of the model and has the same weights of each input word. Finally, the label of input text can be predicted by the last output of the hidden layer.

To diminish the time complexity of the model and capture contextual information, Liu et al. [48] introduce a model for catching the semantics of long texts. It parses the text one by one and is a biased model, making the following inputs profit over the former and decreasing the semantic efficiency of capturing the whole text. For modeling topic labeling tasks with long input sequences, TopicRNN [57] is proposed. It captures the dependencies of words in a document via latent topics and uses RNNs to capture local dependencies and latent topic models for capturing global semantic dependencies. Virtual Adversarial Training (VAT) [141] is a useful regularization method applicable to semi-supervised learning tasks. Miyato et al. [55] apply adversarial and virtual adversarial training to the text field and employ the perturbation into word embedding rather than the original input text. The model improves the quality of the word embedding and is not easy to overfit during training. Capsule network [142] captures the relationships between features using dynamic routing between capsules comprised of a group of neurons in a layer. Wang et al. [119]

propose an RNN-Capsule model with a simple capsule structure for the sentiment classification task.
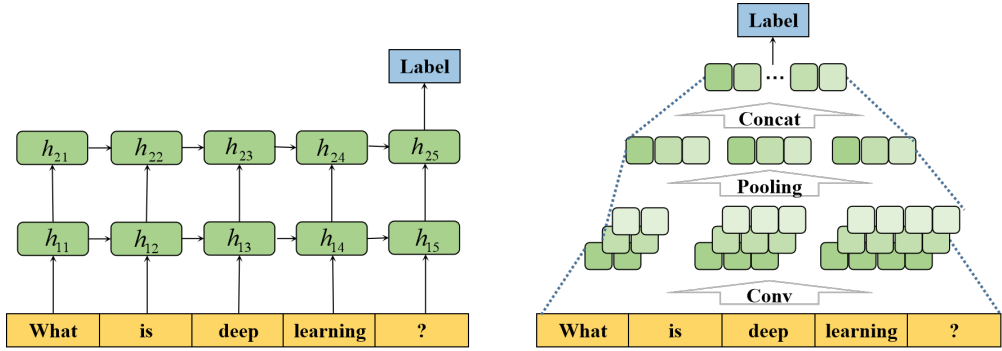

Fig. 7. The RNN based model (left) and the CNN based model (right).

In the backpropagation process of RNN, the weights are adjusted by gradients, calculated by continuous multiplications of derivatives. If the derivatives are extremely small, it may cause a gradient vanishing problem by continuous multiplications. Long Short-Term Memory (LSTM) [143], the improvement of RNN, effectively alleviates the gradient vanishing problem. It is composed of a cell to remember values on arbitrary time intervals and three gate structures to control information flow. The gate structures include input gates, forget gates, and output gates. The LSTM classification method can better capture the connection among context feature words, and use the forgotten gate structure to filter useless information, which is conducive to improving the total capturing ability of the classifier. Tree-LSTM [2] extends the sequence of LSTM models to the tree structure. The whole subtree with little influence on the result can be forgotten through the LSTM forgetting gate mechanism for the Tree-LSTM model.

Natural Language Inference (NLI) predicts whether one text's meaning can be deduced from another by measuring the semantic similarity between each pair of sentences. To consider other granular matchings and matchings in the reverse direction, Wang et al. [144] propose a model for the NLI task named Bilateral multi-perspective matching (BiMPM). It encodes input sentences by the BiLSTM encoder. Then, the encoded sentences are matched in two directions. The results are aggregated in a fixed-length matching vector by another BiLSTM layer. Finally, the result is evaluated by a fully connected layer.

2.2.4 CNN-based Methods. Convolutional neural networks (CNNs) are proposed for image classification with convolving filters that can extract features of pictures. Unlike RNN, CNN can simultaneously apply convolutions defined by different kernels to multiple chunks of a sequence. Therefore, CNNs are used for many NLP tasks, including text classification. For text classification, the text requires being represented as a vector similar to the image representation, and text features can be filtered from multiple angles, as shown in Fig. 7. Firstly, the word vectors of the input text are spliced into a matrix. The matrix is then fed into the convolutional layer, which contains several filters with different dimensions. Finally, the result of the convolutional layer goes through the pooling layer and concatenates the pooling result to obtain the final vector representation of the text. The category is predicted by the final vector.

To try using CNN for the text classification task, an unbiased model of convolutional neural networks is introduced by Kim, called TextCNN [18]. It can better determine discriminative phrases

in the max-pooling layer with one layer of convolution and learn hyperparameters except for word vectors by keeping word vectors static. Training only on labeled data is not enough for data-driven deep models. Therefore, some researchers consider utilizing unlabeled data. Johnson et al. [145] propose a text classification CNN model based on two-view semi-supervised learning, which first uses unlabeled data to train the embedding of text regions and then labeled data. DNNs usually have better performance, but it increases the computational complexity. Motivated by this, a deep pyramid convolutional neural network (DPCNN) [59] is proposed, with a little more computational accuracy, increasing by raising the network depth. The DPCNN is more specific than ResNet [146], as all the shortcuts are exactly simple identity mappings without any complication for dimension matching.

According to the minimum embedding unit of text, embedding methods are divided into character-level, word-level, and sentence-level embedding. Character-level embeddings can settle Out-of-Vocabulary (OOV) words. Word-level embeddings learn the syntax and semantics of the words. Moreover, sentence-level embedding can capture relationships among sentences. Motivated by these, Nguyen et al. [147] propose a deep learning method based on a dictionary, increasing information for word-level embeddings through constructing semantic rules and deep CNN for character-level embeddings. Adams et al. [122] propose a character-level CNN model, called MGTC, to classify multi-lingual texts written. TransCap [148] is proposed to encapsulate the sentence-level semantic representations into semantic capsules and transfer document-level knowledge.

RNN based models capture the sequential information to learn the dependency among input words, and CNN based models extract the relevant features from the convolution kernels. Thus some works study the fusion of the two methods. BLSTM-2DCNN [118] integrates a Bidirectional LSTM (BiLSTM) with two-dimensional max pooling. It uses a 2D convolution to sample more meaningful information of the matrix and understands the context better through BiLSTM. Moreover, Xue et al. [149] propose MTNA, a combination of BiLSTM and CNN layers, to solve aspect category classification and aspect term extraction tasks.
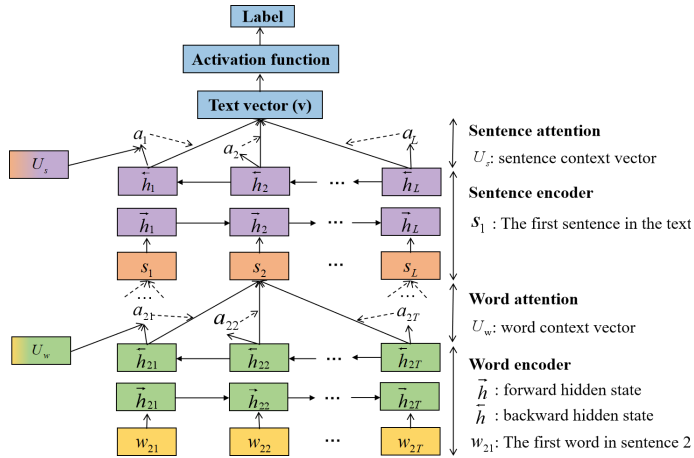

Fig. 8. Hierarchical Attention Network [46].

2.2.5 Attention-based Methods. CNN and RNN provide excellent results on tasks related to text classification. However, these models are not intuitive enough for poor interpretability, especially in classification errors, which cannot be explained due to the non-readability of hidden data. The

attention-based methods are successfully used in the text classification. Bahdanau et al. [150] first propose an attention mechanism that can be used in machine translation. Motivated by this, Yang et al. [46] introduce the hierarchical attention network (HAN) to gain better visualization by employing the extremely informational components of a text, as shown in Fig. 8. HAN includes two encoders and two levels of attention layers. The attention mechanism lets the model pay different attention to specific inputs. It aggregates essential words into sentence vectors firstly and then aggregates vital sentence vectors into text vectors. It can learn how much contribution of each word and sentence for the classification judgment, which is beneficial for applications and analysis through the two levels of attention.

The attention mechanism can improve the performance with interpretability for text classification, which makes it popular. There are some other works based on attention. LSTMN [50] is proposed to process text step by step from left to right and does superficial reasoning through memory and attention. Wang et al. [151] propose an attention-based LSTM neural network by exploring the connection between the aspects and the input sentences. BI-Attention [125] is proposed for cross-lingual text classification to catch bilingual long-distance dependencies. Hu et al. [152] propose an attention mechanism based on category attributes for solving the imbalance of the number of various charges which contain few-shot charges.

Self-attention [153] captures the weight distribution of words in sentences by constructing K, Q and V matrices among sentences that can capture long-range dependencies on text classification. We give an example for self-attention, as shown in Fig. 9. Each input word vector a_i can be represented as three n-dimensional vectors, including q_i, k_i and v_i. After self-attention, the output vector b_i can be represented as $\sum_j softmax(a_{ij})v_j$ and $a_{ij} = q_i \cdot k_j/\sqrt{n}$. All output vectors can be parallelly computed. Lin et al. [130] used source token self-attention to explore the weight of every token to the entire sentence in the sentence representation task. To capture long-range dependencies, Bi-directional Block Self-Attention Network (Bi-BloSAN) [133] uses an intra-block self-attention network (SAN) to every block split by sequence and an inter-block SAN to the outputs.
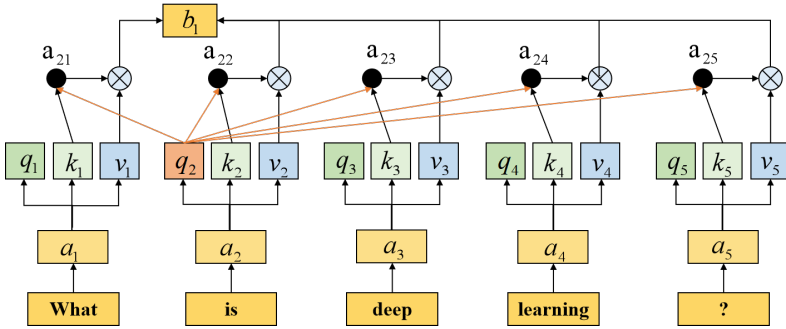
Fig. 9. An example of self-attention.

Aspect-based sentiment analysis (ABSA) breaks down a text into multiple aspects and allocates each aspect a sentiment polarity. The sentiment polarity can be divided into two types: positive, neutral and negative. Some attention-based models are proposed to identify the fine-grained opinion polarity towards a specific aspect for aspect-based sentiment tasks. ATAE-LSTM [151] can concentrate on different parts of each sentence according to the input through the attention mechanisms. MGAN [80] presents a fine-grained attention mechanism with a coarse-grained attention mechanism to learn the word-level interaction between context and aspect.

To catch the complicated semantic relationship among each question and candidate answers for the QA task, Tan et al. [154] introduce CNN and RNN and generate answer embeddings by using a simple one-way attention mechanism affected through the question context. The attention captures the dependence among the embeddings of questions and answers. Extractive QA can be seen as the text classification task. It inputs a question and multiple candidates answers and classifies every candidate answer to recognize the correct answer. Furthermore, AP-BILSTM [127] with a two-way attention mechanism can learn the weights between the question and each candidate answer to obtain the importance of each candidate answer to the question.

2.2.6 Transformer-based Methods. Pre-trained language models effectively learn global semantic representation and significantly boost NLP tasks, including text classification. It generally uses unsupervised methods to mine semantic knowledge automatically and then construct pre-training targets so that machines can learn to understand semantics.

As shown in Fig. 10, we give differences in the model architectures among ELMo [66], OpenAI GPT [155], and BERT [19]. ELMo [66] is a deep contextualized word representation model, which is readily integrated into models. It can model complicated characteristics of words and learn different representations for various linguistic contexts. It learns each word embedding according to the context words with the bi-directional LSTM. GPT [155] employs supervised fine-tuning and unsupervised pre-training to learn general representations that transfer with limited adaptation to many NLP tasks. Furthermore, the domain of the target task does not need to be similar to the unlabeled datasets. The training procedure of the GPT algorithm usually includes two stages. Firstly, the initial parameters of a neural network model are learned by a modeling objective on the unlabeled dataset. We can then employ the corresponding supervised objective to accommodate these parameters for the target task. To pre-train deep bidirectional representations from the unlabeled text through joint conditioning on both left and right context in every layer, BERT model [19], proposed by Google, significantly improves performance on NLP tasks, including text classification. It is fine-tuned by adding just an additional output layer to construct models for multiple NLP tasks, such as SA, QA, and machine translation. Comparing with these three models, ELMo is a feature-based method using LSTM, and BERT and OpenAI GPT are fine-tuning approaches using Transformer. Furthermore, ELMo and BERT are bidirectional training models and OpenAI GPT is training from left to right. Therefore, BERT gets a better result, which combines the advantages of ELMo and OpenAI GPT.
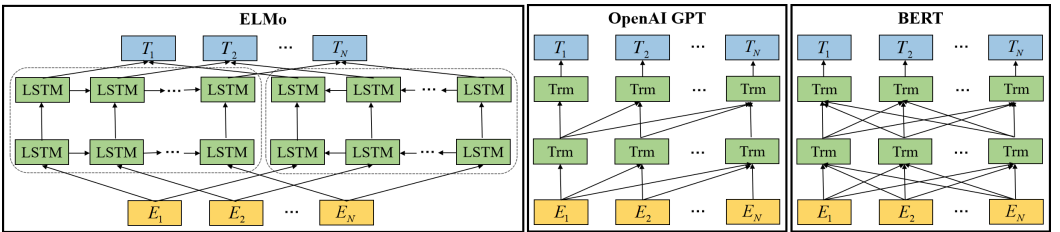
Fig. 10. Differences in pre-trained model architectures [19], including BERT, OpenAI GPT and ELMo. E_i represents embedding of i th input. Trm represents the transformer block. T_i represents predicted tag of i th input.

Transformer-based models can parallelize computation without considering the sequential information suitable for large scale datasets, making it popular for NLP tasks. Thus, some other works are used for text classification tasks and get excellent performance. RoBERTa [87] adopts

the dynamic masking method that generates the masking pattern every time with a sequence to be fed into the model. It uses more data for longer pre-training and estimates the influence of various essential hyperparameters and the size of training data. ALBERT [89] uses two-parameter simplification schemes. In general, these methods adopt unsupervised objective functions for pre-training, including the next sentence prediction, masking technology, and permutation. These target functions based on the word prediction demonstrate a strong ability to learn the word dependence and semantic structure [156]. XLNet [85] is a generalized autoregressive pre-training approach. It maximizes the expected likelihood across the whole factorization order permutations to learn the bidirectional context. Furthermore, it can overcome the weaknesses of BERT by an autoregressive formulation and integrate ideas from Transformer-XL [157] into pre-training.

2.2.7 GNN-based Methods. The DNN models like CNN get great performance on regular structure, not for arbitrarily structured graphs. Some researchers study how to expand on arbitrarily structured graphs [158] [159]. With the increasing attention of graph neural networks (GNNs), GNN-based models obtain excellent performance by encoding syntactic structure of sentences on semantic role labeling task [160], relation classification task [161] and machine translation task [162]. It turns text classification into a graph node classification task. We show a GCN model for text classification with four input texts, as shown in Fig. 11. Firstly, the four input texts and the words in the text, defined as nodes, are constructed into the graph structures. The graph nodes are connected by bold black edges, which indicates document-word edges and word-word edges. The weight of each word-word edge usually means their co-occurrence frequency in the corpus. Then, the words and texts are represented through the hidden layer. Finally, the label of all input texts can be predicted by the graph.

The GNN-based models can learn the syntactic structure of sentences making some researchers study using GNN for text classification. DGCNN [68] is a graph-CNN converting text to graph-of-words, having the advantage of learning different levels of semantics with CNN models. Yao et al. [6] propose the text graph convolutional network (TextGCN), which builds a heterogeneous word text graph for a whole dataset and captures global word co-occurrence information. To enable GNN-based models to underpin online testing, Huang et al. [137] build graphs for each text with global parameter sharing, not a corpus-level graph structure, to help preserve global information and reduce the burden. TextING [163] builds individual graphs for each document and learns text-level word interactions by GNN to effectively produce embeddings for obscure words in the new text.
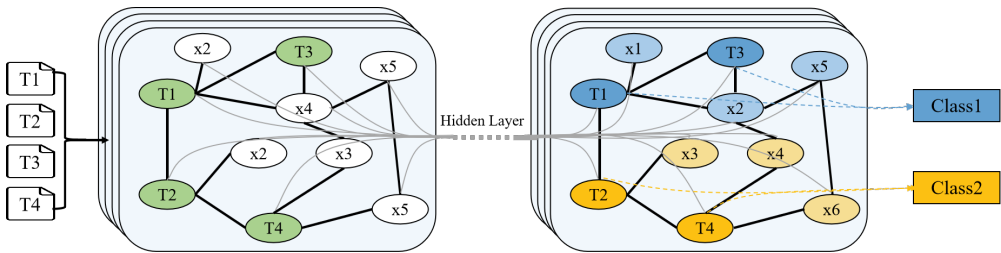

Fig. 11. The GCN based model. Black bold edges are document-word edges and word-word edges in the graph.

Graph attention networks (GATs) [164] employ masked self-attention layers by attending over its neighbors. Thus, some GAT-based models are proposed to compute the hidden representations of each node. The heterogeneous graph attention network (HGAT) [165] with a dual-level attention

mechanism learns the importance of different neighboring nodes and node types in the current node. The model propagates information on the graph and captures the relations to address the semantic sparsity for semi-supervised short text classification. MAGNET [139] is proposed to capture the correlation among the labels based on GATs, which learns the crucial dependencies between the labels and generates classifiers by a feature matrix and a correlation matrix.

Event prediction (EP) can be divided into generated event prediction and selective event prediction (also known as script event prediction). EP, referring to scripted event prediction in this review, infers the subsequent event according to the existing event context. Unlike other text classification tasks, texts in EP are composed of a series of sequential subevents. Extracting features of the relationship among such subevents is of critical importance. SGNN [76] is proposed to model event interactions and learn better event representations by constructing an event graph to utilize the event network information better. The model makes full use of dense event connections for the EP task.

2.2.8 Others. In addition to all the above models, there are some other individual models. Here we introduce some exciting models.

Siamese neural network. The siamese neural network [166] is also called a twin neural network (Twin NN). It utilizes equal weights while working in tandem using two distinct input vectors to calculate comparable output vectors. Mueller et al. [167] present a siamese adaptation of the LSTM network comprised of couples of variable-length sequences. The model is employed to estimate the semantic similarity among texts, exceeding carefully handcrafted features and proposed neural network models of higher complexity. The model further represents text employing neural networks whose inputs are word vectors learned separately from a vast dataset. To settle unbalanced data classification in the medical domain, Jayadeva et al. [168] use a Twin NN model to learn from enormous unbalanced corpora. The objective functions achieve the Twin SVM approach with non-parallel decision boundaries for the corresponding classes, and decrease the Twin NN complexity, optimizing the feature map to better discriminate among classes.

Virtual adversarial training (VAT). Deep learning methods require many extra hyperparameters, which increase the computational complexity. VAT [169], regularization based on local distributional smoothness can be used in semi-supervised tasks, requires only a small number of hyperparameters, and can be interpreted directly as robust optimization. Miyato et al. [55] use VAT to effectively improve the robustness and generalization ability of the model and word embedding performance.

Reinforcement learning (RL). RL learns the best action in a given environment through maximizing cumulative rewards. Zhang et al. [170] offer an RL approach to establish structured sentence representations via learning the structures related to tasks. The model has Information Distilled LSTM (ID-LSTM) and Hierarchical Structured LSTM (HS-LSTM) representation models. The ID-LSTM learns the sentence representation by choosing essential words relevant to tasks, and the HS-LSTM is a two-level LSTM for modeling sentence representation.

QA style for the sentiment classification task. It is an interesting attempt to treat the sentiment classification task as a QA task. Shen et al. [171] create a high-quality annotated corpus. A three-stage hierarchical matching network was proposed to consider the matching information between questions and answers.

External commonsense knowledge. Due to the insufficient information of the event itself to distinguish the event for the EP task, Ding et al. [172] consider that the event extracted from the original text lacked common knowledge, such as the intention and emotion of the event participants. The model improves the effect of stock prediction, EP, and so on.

Quantum language model. In the quantum language model, the words and dependencies among words are represented through fundamental quantum events. Zhang et al. [173] design a quantum-inspired sentiment representation method to learn both the semantic and the sentiment information of subjective text. By inputting density matrices to the embedding layer, the performance of the model improves.

Summary. Deep Learning consists of multiple hidden layers in a neural network with a higher level of complexity and can be trained on unstructured data. Deep learning architecture can learn feature representations directly from the input without too many manual interventions and prior knowledge. However, deep learning technology is a data-driven method, which usually needs enormous data to achieve high performance. Although self-attention based models can bring some interpretability among words for DNNs, it is not enough comparing with shallow models to explain why and how it works well.

3 DATASETS AND EVALUATION METRICS

3.1 Datasets

The availability of labeled datasets for text classification has become the main driving force behind the fast advancement of this research field. In this section, we summarize the characteristics of these datasets in terms of domains and give an overview in Table 3, including the number of categories, average sentence length, the size of each dataset, related papers, data sources to access and applications.

Sentiment Analysis (SA). SA is the process of analyzing and reasoning the subjective text within emotional color. It is crucial to get information on whether it supports a particular point of view from the text that is distinct from the traditional text classification that analyzes the objective content of the text. SA can be binary or multi-class. Binary SA is to divide the text into two categories, including positive and negative. Multi-class SA classifies text to multi-level or fine-grained labels. The SA datasets include MR, SST, MPQA, IMDB, Yelp, AM, Subj [177], CR [179], SS-Twitter, SS-Youtube, Twitter, SE1604, EP and so on. Here we detail several of the primary datasets.

Movie Review (MR). The MR [207] [174] is a movie review dataset, each of which corresponds to a sentence. The corpus has 5,331 positive data and 5,331 negative data. 10-fold cross-validation by random splitting is commonly used to test MR.

Stanford Sentiment Treebank (SST). The SST [175] is an extension of MR. It has two categories. SST-1 with fine-grained labels with five classes. It has 8,544 training texts and 2,210 test texts, respectively. Furthermore, SST-2 has 9,613 texts with binary labels being partitioned into 6,920 training texts, 872 development texts, and 1,821 testing texts.

The Multi-Perspective Question Answering (MPQA). The MPQA [208] [180] is an opinion dataset. It has two class labels and also an MPQA dataset of opinion polarity detection sub-tasks. MPQA includes 10,606 sentences extracted from news articles from various news sources. It should be noted that it contains 3,311 positive texts and 7,293 negative texts without labels of each text.

IMDB reviews. The IMDB review [183] is developed for binary sentiment classification of film reviews with the same amount in each class. It can be separated into training and test groups on average, by 25,000 comments per group.

Yelp reviews. The Yelp review [189] is summarized from the Yelp Dataset Challenges in 2013, 2014, and 2015. This dataset has two categories. Yelp-2 of these were used for negative and positive emotion classification tasks, including 560,000 training texts and 38,000 test texts. Yelp-5 is used to detect fine-grained affective labels with 650,000 training and 50,000 test texts in all classes.

Amazon Reviews (AM). The AM [5] is a popular corpus formed by collecting Amazon website product reviews [190]. This dataset has two categories. The Amazon-2 with two classes includes

Table 3. Summary statistics for the datasets. C: Number of target classes. L: Average sentence length. N: Dataset size.

Datasets	C	L	N	Related Papers	Sources	Applications
MR	2	20	10,662	[18] [7] [78] [6]	[174]	SA
SST-1	5	18	11,855	[33] [18] [2] [3][50]	[175]	SA
SST-2	2	19	9,613	[33] [18] [8] [48] [19]	[176]	SA
Subj	2	23	10,000	[18] [48] [78]	[177]	QA
TREC	6	10	5,952	[18] [7] [8] [121]	[178]	QA
CR	2	19	3,775	[18] [78]	[179]	QA
MPQA	2	3	10,606	[29] [18] [133]	[180]	SA
Twitter	3	19	11,209	[7][121]	[181]	SA
EP	5	129	31,675	[29]	[182]	SA
IMDB	2	294	50,000	[35] [41] [8] [48] [55] [85]	[183]	SA
20NG	20	221	18,846	[39] [117] [124] [6] [136]	[184]	NC
Fudan	20	2981	18,655	[39]	[185]	TL
AG News	4	45/7	127,600	[5] [59] [121] [78] [85]	[186]	NC
Sogou	6	578	510,000	[5]	[187]	NC
DBPedia	14	55	630,000	[5] [59] [55] [135]	[188]	TL
Yelp.P	2	153	598,000	[5] [59]	[189]	SA
Yelp.F	5	155	700,000	[5] [46] [59]	[189]	SA
YahooA	10	112	1,460,000	[5] [46]	[5]	TL
Amz.P	2	91	4,000,000	[131] [5]	[190]	SA
Amz.F	5	93	3,650,000	[5] [46] [131]	[190]	SA
DSTC 4	89	-	30,000	[9]	[191]	DAC
MRDA	5	-	62,000	[9]	[192]	DAC
SwDA	43	-	1,022,000	[9]	[193]	DAC
RCV1	103	240	807,595	[117] [123] [68] [134] [139]	[194]	NC
RCV1-V2	103	124	804,414	[74] [139]	[195]	NC
NLP&CC 2013	2	-	115,606	[125]	[126]	SA
SS-Twitter	2	-	2,113	[63]	[196]	SA
SS-Youtube	2	-	2,142	[63]	[196]	SA
SE1604	3	-	39,141	[63]	[197]	SA
Bing	4	20	34,871	[121]	[198]	TL
AAPD	54	163	55,840	[74] [139]	[75]	TL
Reuters	90	1	10,788	[78] [139]	[199]	NC
R8	8	66	7,674	[6] [136] [137]	[200]	NC
R52	52	70	9,100	[6] [136] [137]	[200]	NC
NYTimes	2,318	629	1,855,659	[68]	[201]	NC
SQuAD	-	5,000	5,570	[66] [66] [87] [89]	[202]	QA
WikiQA	-	873	243	[127]	[203]	QA
Ohsumed	23	136	7,400	[6] [136] [137]	[204]	TL
Amazon670K	670	244	643,474	[123] [131]	[205]	TL
EUR-Lex	3,956	1,239	19,314	[120] [131] [134] [138] [134]	[206]	TL

3,600,000 training sets and 400,000 testing sets. Amazon-5, with five classes, includes 3,000,000 and 650,000 comments for training and testing.

News Classification (NC). News content is one of the most crucial information sources which has a critical influence on people. The NC system facilitates users to get vital knowledge in real-time. News classification applications mainly encompass: recognizing news topics and recommending related news according to user interest. The news classification datasets include 20NG, AG, R8, R52, Sogou, and so on. Here we detail several of the primary datasets.

20 Newsgroups (20NG). The 20NG [184] is a newsgroup text dataset. It has 20 categories with the same number of each category and includes 18,846 texts.

AG News (AG). The AG News [5] [186] is a search engine for news from academia, choosing the four largest classes. It uses the title and description fields of each news. AG contains 120,000 texts for training and 7,600 texts for testing.

R8 and R52. R8 and R52 are two subsets which are the subset of Reuters [199]. R8 [200] has 8 categories, divided into 2,189 test files and 5,485 training courses. R52 has 52 categories, split into 6,532 training files and 2,568 test files.

Sogou News (Sogou). The Sogou News [135] combines two datasets, including SogouCA and SogouCS news sets. The label of each text is the domain names in the URL.

Topic Labeling (TL). The topic analysis attempts to get the meaning of the text by defining the sophisticated text theme. The topic labeling is one of the essential components of the topic analysis technique, intending to assign one or more subjects for each document to simplify the topic analysis. The topic labeling datasets include DBPedia, Ohsumed, EUR-Lex, WOS, PubMed, and YahooA. Here we detail several of the primary datasets.

DBpedia. The DBpedia [188] is a large-scale multi-lingual knowledge base generated using Wikipedia's most ordinarily used infoboxes. It publishes DBpedia each month, adding or deleting classes and properties in every version. DBpedia's most prevalent version has 14 classes and is divided into 560,000 training data and 70,000 test data.

Ohsumed. The Ohsumed [204] belongs to the MEDLINE database. It includes 7,400 texts and has 23 cardiovascular disease categories. All texts are medical abstracts and are labeled into one or more classes.

Yahoo answers (YahooA). The YahooA [5] is a topic labeling task with 10 classes. It includes 140,000 training data and 5,000 test data. All text contains three elements, being question titles, question contexts, and best answers, respectively.

Question Answering (QA). The QA task can be divided into two types: the extractive QA and the generative QA. The extractive QA gives multiple candidate answers for each question to choose which one is the right answer. Thus, the text classification models can be used for the extractive QA task. The QA discussed in this paper is all extractive QA. The QA system can apply the text classification model to recognize the correct answer and set others as candidates. The question answering datasets include SQuAD, MS MARCO, TREC-QA, WikiQA, and Quora [209]. Here we detail several of the primary datasets.

Stanford Question Answering Dataset (SQuAD). The SQuAD [202] is a set of question and answer pairs obtained from Wikipedia articles. The SQuAD has two categories. SQuAD1.1 contains 536 pairs of 107,785 Q&A items. SQuAD2.0 combines 100,000 questions in SQuAD1.1 with more than 50,000 unanswerable questions that crowd workers face in a form similar to answerable questions [210].

MS MARCO. The MS MARCO [211] contains questions and answers. The questions and part of the answers are sampled from actual web texts by the Bing search engine. Others are generative. It is used for developing generative QA systems released by Microsoft.

TREC-QA. The TREC-QA [178] includes 5,452 training texts and 500 testing texts. It has two versions. TREC-6 contains 6 categories, and TREC-50 has 50 categories.

WikiQA. The WikiQA dataset [203] includes questions with no correct answer, which needs to evaluate the answer.

Natural Language Inference (NLI). NLI is used to predict whether the meaning of one text can be deduced from another. Paraphrasing is a generalized form of NLI. It uses the task of measuring the semantic similarity of sentence pairs to decide whether one sentence is the interpretation of another. The NLI datasets include SNLI, MNLI, SICK, STS, RTE, SciTail, MSRP, etc. Here we detail several of the primary datasets.

The Stanford Natural Language Inference (SNLI). The SNLI [212] is generally applied to NLI tasks. It contains 570,152 human-annotated sentence pairs, including training, development, and test sets, which are annotated with three categories: neutral, entailment, and contradiction.

Multi-Genre Natural Language Inference (MNLI). The Multi-NLI [213] is an expansion of SNLI, embracing a broader scope of written and spoken text genres. It includes 433,000 sentence pairs annotated by textual entailment labels.

Sentences Involving Compositional Knowledge (SICK). The SICK [214] contains almost 10,000 English sentence pairs. It consists of neutral, entailment and contradictory labels.

Microsoft Research Paraphrase (MSRP). The MSRP [215] consists of sentence pairs, usually for the text-similarity task. Each pair is annotated by a binary label to discriminate whether they are paraphrases. It respectively includes 1,725 training and 4,076 test sets.

Dialog Act Classification (DAC). A dialog act describes an utterance in a dialog based on semantic, pragmatic, and syntactic criteria. DAC labels a piece of a dialog according to its category of meaning and helps learn the speaker's intentions. It is to give a label according to dialog. Here we detail several of the primary datasets, including DSTC 4, MRDA, and SwDA.

Dialog State Tracking Challenge 4 (DSTC 4). The DSTC 4 [191] is used for dialog act classification. It has 89 training classes, 24,000 training texts, and 6,000 testing texts.

ICSI Meeting Recorder Dialog Act (MRDA). The MRDA [192] is used for dialog act classification. It has 5 training classes, 51,000 training texts, 11,000 testing texts, and 11,000 validation texts.

Switchboard Dialog Act (SwDA). The SwDA [193] is used for dialog act classification. It has 43 training classes, 1,003,000 training texts, 19,000 testing texts and 112,000 validation texts.

Multi-label datasets. In multi-label classification, an instance has multiple labels, and each label can only take one of the multiple classes. There are many datasets based on multi-label text classification. It includes Reuters, Education, Patent, RCV1, RCV1-2K, AmazonCat-13K, BlurbGenreCollection, WOS-11967, AAPD, etc. Here we detail several of the main datasets.

Reuters news. The Reuters [200] [199] is a popularly used dataset for text classification from Reuters financial news services. It has 90 training classes, 7,769 training texts, and 3,019 testing texts, containing multiple labels and single labels. There are also some Reuters sub-sets of data, such as R8, BR52, RCV1, and RCV1-v2.

Patent Dataset. The Patent Dataset is obtained from USPTO [1], which is a patent system grating U.S. patents containing textual details such title and abstract. It contains 100,000 US patents awarded in the real-world with multiple hierarchical categories.

Reuters Corpus Volume I (RCV1) and RCV1-2K. The RCV1 [194] is collected from Reuters News articles from 1996-1997, which is human-labeled with 103 categories. It consists of 23,149 training and 784,446 testing texts, respectively. The RCV1-2K dataset has the same features as the

---

[1]https://www.uspto.gov/

Table 4. The notations used in evaluation metrics.

Notations	Descriptions
TP	true positive
FP	false positive
TN	true negative
FN	false negative
TP_t	true positive of the t th label on a text
FP_t	false positive of the t th label on a text
TN_t	true negative of the t th label on a text
FN_t	false negative of the t th label on a text
$\mathcal{S}$	label set of all samples
$\mathcal{L}$	the number of ground truth labels or possible answers on each text
Q	the number of predicted labels on each text
$rank_i$	the ranking of the ground-truth answer at answer i th
k	the number of selected labels on extreme multi-label text classification

RCV1. However, the label set of RCV1-2K has been expanded with some new labels. It contains 2456 labels.

Web of Science (WOS-11967). The WOS-11967 [216] is crawled from the Web of Science, consisting of abstracts of published papers with two labels for each example. It is shallower, but significantly broader, with fewer classes in total.

Arxiv Academic Paper Dataset (AAPD). The AAPD [75] is a large dataset in the computer science field for the multi-label text classification from website [2]. It has 55,840 papers, including the abstract and the corresponding subjects with 54 labels in total. The aim is to predict the corresponding subjects of each paper according to the abstract.

Others. There are some datasets for other applications, such as Geonames toponyms, Twitter posts, and so on.

3.2 Evaluation Metrics

In terms of evaluating text classification models, accuracy and F1 score are the most used to assess the text classification methods. Later, with the increasing difficulty of classification tasks or the existence of some particular tasks, the evaluation metrics are improved. For example, evaluation metrics such as P@K and Micro-F1 are used to evaluate multi-label text classification performance, and MRR is usually used to estimate the performance of QA tasks. In Table 4, we give the notations used in evaluation metrics.

3.2.1 Single-label metrics. Single-label text classification divides the text into one of the most likely categories applied in NLP tasks such as QA, SA, and dialogue systems [9]. For single-label text classification, one text belongs to just one catalog, making it possible not to consider the relations among labels. Here we introduce some evaluation metrics used for single-label text classification tasks.

Accuracy and Error Rate. Accuracy and Error Rate are the fundamental metrics for a text classification model. The Accuracy and Error Rate are respectively defined as

$$Accuracy = \frac{(TP + TN)}{N}, \quad ErrorRate = 1 - Accuracy = \frac{(FP + FN)}{N}.$$

---

[2]https://arxiv.org/

Precision, Recall and F1. These are vital metrics utilized for unbalanced test sets regardless of the standard type and error rate. For example, most of the test samples have a class label. F1 is the harmonic average of Precision and Recall. Accuracy, Recall, and F1 as defined

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}, \quad F1 = \frac{2 \, Precision \times Recall}{Precision \, + Recall}.$$

The desired results will be obtained when the accuracy, F1 and recall value reach 1. On the contrary, when the values become 0, the worst result is obtained. For the multi-class classification problem, the precision and recall value of each class can be calculated separately, and then the performance of the individual and whole can be analyzed.

Exact Match (EM). The EM is a metric for QA tasks measuring the prediction that matches all the ground-truth answers precisely. It is the primary metric utilized on the SQuAD dataset.

Mean Reciprocal Rank (MRR). The MRR is usually applied for assessing the performance of ranking algorithms on QA and Information Retrieval (IR) tasks. MRR is defined as MRR = $\frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i}$.

Hamming-loss (HL). The HL [217] assesses the score of misclassified instance-label pairs where a related label is omitted or an unrelated is predicted.

3.2.2 Multi-label metrics. Compared with single-label text classification, multi-label text classification divides the text into multiple category labels, and the number of category labels is variable. These metrics are designed for single label text classification, which are not suitable for multi-label tasks. Thus, there are some metrics designed for multi-label text classification.

Micro − F1. The $Micro − F1$ [218] is a measure that considers the overall accuracy and recall of all labels. The $Micro − F1$ is defined as:

$$Micro − F1 = \frac{2P_t \times R_t}{P + R},$$

where:

$$P = \frac{\sum_{t \in S} TP_t}{\sum_{t \in S} TP_t + FP_t}, \quad R = \frac{\sum_{t \in S} TP_t}{\sum_{t \in S} TP_t + FN_t}.$$

Macro − F1. The $Macro − F1$ calculates the average $F1$ of all labels. Unlike $Micro − F_1$, which sets even weight to every example, $Macro − F1$ sets the same weight to all labels in the average process. Formally, $Macro − F1$ is defined as:

$$Macro − F1 = \frac{1}{S} \sum_{t \in S} \frac{2P_t \times R_t}{P_t + R_t},$$

where:

$$P_t = \frac{TP_t}{TP_t + FP_t}, \quad R_t = \frac{TP_t}{TP_t + FN_t}.$$

In addition to the above evaluation metrics, there are some rank-based evaluation metrics for extreme multi-label classification tasks, including P@K and NDCG@K.

Precision at Top K (P@K). The $P@K$ is the precision at the top k. For $P@K$, each text has a set of $\mathcal{L}$ ground truth labels $L_t = \{l_0, l_1, l_2 \ldots, l_{\mathcal{L}-1}\}$, in order of decreasing probability $P_t = [p_0, p_1, p_2 \ldots, p_{Q-1}]$. The precision at k is

$$P@K = \frac{1}{k} \sum_{j=0}^{\min(\mathcal{L}, k)-1} \text{rel}_{L_i}\left(P_t(j)\right),$$

where

$$\mathrm{rel}_L(p) = \left\{ \begin{array}{ll} 1 & \text{if } p \in L \\ 0 & \text{otherwise} \end{array} \right. .$$

Normalized Discounted Cummulated Gains (NDCG@K). The NDCG at k is

$$\mathrm{NDCG@}K = \frac{1}{\mathrm{IDCG}\,(\mathrm{L}_i, k)} \sum_{j=0}^{n-1} \frac{rel_{L_i}\,(P_t(j))}{\ln(j+1)},$$

where

$$n = \min\left(\max\left(|P_i|, |L_i|\right), k\right).$$

4 QUANTITATIVE RESULTS

In this section, we tabulate the performance of the main models on classic datasets evaluated by classification accuracy, as shown in Table 5, including MR, SST-2, IMDB, Yelp.P, Yelp.F, Amazon.F, 20NG, AG, DBpedia and SNLI. We can see that BERT based models get better results on most datasets, which means that if you need to implement a text classification task, you can try BERT based models firstly, except MR and 20NG, which have not been experimented on BERT based models. RNN-Capsule [119] obtains the best result on MR and BLSTM-2DCNN [118] gets the best on 20NG.

5 FUTURE RESEARCH CHALLENGES

Text classification – as efficient information retrieval and mining technology – plays a vital role in managing text data. It uses NLP, data mining, machine learning, and other techniques to automatically classify and discover different text types. Text classification takes multiple types of text as input, and the text is represented as a vector by the pre-training model. Then the vector is fed into the DNN for training until the termination condition is reached, and finally, the performance of the training model is verified by the downstream task. Existing models have already shown their usefulness in text classification, but there are still many possible improvements to explore.

Although some new text classification models repeatedly brush up the accuracy index of most classification tasks, it cannot indicate whether the model "understands" the text from the semantic level like human beings. Moreover, with the emergence of the noise sample, the small sample noise may cause the decision confidence to change substantially or even lead to decision reversal. Therefore, the semantic representation ability and robustness of the model need to be proved in practice. Besides, the pre-trained semantic representation model represented by word vectors can often improve the performance of downstream NLP tasks. The existing research on the transfer strategy of context-free word vectors is still relatively preliminary. Thus, we conclude from data, models, and performance perspective, text classification mainly faces the following challenges.

5.1 Data

For a text classification task, data is essential to model performance, whether it is shallow learning or deep learning method. The text data mainly studied includes multi-chapter, short text, cross-language, multi-label, less sample text. For the characteristics of these data, the existing technical challenges are as follows:

Zero-shot/Few-shot learning. The current model of deep learning is too dependent on numerous labeled data. The performance of these models is significantly affected in zero-shot or few-shot learning.

The external knowledge. As we all know, the more beneficial information is input into a DNN, its better performance. Therefore, we believe that adding external knowledge (knowledge base or

Table 5. Accuracy of deep learning-based text classification models on primary datasets evaluated by classification accuracy (in terms of publication year). Bold is the most accurate.

	Sentiment						News		Topic	NLI
Model	MR	SST-2	IMDB	Yelp.P	Yelp.F	Amz.F	20NG	AG	DBpedia	SNLI
RAE [29]	77.7	82.4		-	-	-	-	-	-	-
MV-RNN [31]	79	82.9	-	-	-	-	-	-	-	-
RNTN [33]	75.9	85.4	-	-	-	-	-	-	-	-
DCNN [7]		86.8	89.4	-	-	-	-	-	-	-
Paragraph-Vec [35]		87.8	92.58	-	-	-	-	-	-	-
TextCNN[18]	81.5	88.1	-	-	-	-	-	-	-	-
TextRCNN [39]	-	-	-	-	-	-	96.49	-	-	-
DAN [41]	-	86.3	89.4	-	-	-	-	-	-	-
Tree-LSTM [2]		88	-	-	-	-	-	-	-	-
CharCNN [5]	-	-	-	95.12	62.05	-	-	90.49	98.45	-
HAN [46]	-	-	49.4	-	-	63.6	-	-	-	-
SeqTextRCNN [9]	-	-	-	-	-	-	-	-	-	-
oh-2LSTMp [117]	-	-	94.1	97.1	67.61	-	86.68	93.43	99.16	-
LSTMN [50]	-	87.3	-	-	-	-	-	-	-	-
Multi-Task [48]	-	87.9	91.3	-	-	-	-	-	-	-
BLSTM-2DCNN [118]	82.3	89.5	-	-	-	-	**96.5**	-	-	-
TopicRNN [57]	-	-	93.72	-	-	-	-	-	-	-
DPCNN [59]	-	-	-	97.36	69.42	65.19	-	93.13	99.12	-
KPCNN [121]	83.25	-	-	-	-	-	-	88.36	-	-
RAM [65]	-	-	-	-	-	-	-	-	-	-
RNN-Capsule [119]	**83.8**		-	-	-	-	-	-	-	-
ULMFiT [70]	-	-	95.4	97.84	71.02	-	-	94.99	99.2	-
LEAM[72]	76.95	-	-	95.31	64.09	-	81.91	92.45	99.02	-
TextCapsule [78]	82.3	86.8	-	-	-	-	-	92.6	-	-
TextGCN [6]	76.74	-	-	-	-	-	86.34	67.61	-	-
BERT-base [19]	-	93.5	95.63	98.08	70.58	61.6	-	-	-	91.0
BERT-large [19]	-	94.9	95.79	98.19	71.38	62.2	-	-	-	91.7
MT-DNN[83]	-	95.6	83.2	-	-	-	-	-	-	91.5
XLNet-Large [85]	-	96.8	**96.21**	**98.45**	**72.2**	**67.74**	-	-	-	-
XLNet [85]	-	**97**	-	-	-	-	-	**95.51**	**99.38**	-
RoBERTa [87]	-	96.4	-	-	-	-	-	-	-	**92.6**

knowledge graph) is an efficient way to promote the model's performance. Nevertheless, how and what to add is still a challenge.

The multi-label text classification task. Multi-label text classification requires full consideration of the semantic relationship among labels, and the embedding and encoding of the model is a process of lossy compression. Therefore, how to reduce the loss of hierarchical semantics and retain rich and complex document semantic information during training is still a problem to be solved.

Special domain with many terminologies. Texts in a particular field, such as financial and medical texts, contain many specific words or domain experts intelligible slang, abbreviations, etc., which make the existing pre-trained word vectors challenging to work on.

5.2 Models

Most existing structures of shallow and deep learning models are tried for text classification, including integration methods. BERT learns a language representation that can be used to fine-tune

for many NLP tasks. The primary method is to increase data, improve computation power, and design training procedures for getting better results How to tradeoff between data and compute resources and prediction performance is worth studying.

5.3 Performance

The shallow model and the deep model can achieve good performance in most text classification tasks, but the anti-interference ability of their results needs to be improved. How to realize the interpretation of the deep model is also a technical challenge.

The semantic robustness of the model. In recent years, researchers have designed many models to enhance the accuracy of text classification models. However, when there are some adversarial samples in the datasets, the model's performance decreases significantly. Consequently, how to improve the robustness of models is a current research hotspot and challenge.

The interpretability of the model. DNNs have unique advantages in feature extraction and semantic mining and have achieved excellent text classification tasks. However, deep learning is a black-box model, the training process is challenging to reproduce, and the implicit semantics and output interpretability are poor. It makes the improvement and optimization of the model, losing clear guidelines. Furthermore, we cannot accurately explain why the model improves performance.

6 CONCLUSION

This paper principally introduces the existing models for text classification tasks from shallow learning to deep learning. Firstly, we introduce some primary shallow learning models and deep learning models with a summary table. The shallow model improves text classification performance mainly by improving the feature extraction scheme and classifier design. In contrast, the deep learning model enhances performance by improving the presentation learning method, model structure, and additional data and knowledge. Then, we introduce the datasets with a summary table and evaluation metrics for single-label and multi-label tasks. Furthermore, we give the quantitative results of the leading models in a summary table under different applications for classic text classification datasets. Finally, we summarize the possible future research challenges of text classification.

REFERENCES

[1] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pp. 142–150, 2011.

[2] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 1556–1566, 2015.

[3] X. Zhu, P. Sobhani, and H. Guo, "Long short-term memory over recursive structures," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1604–1612, 2015.

[4] S. I. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pp. 90–94, 2012.

[5] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 649–657, 2015.

[6] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 7370–7377, 2019.

[7] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pp. 655–665, 2014.

[8] P. Liu, X. Qiu, X. Chen, S. Wu, and X. Huang, "Multi-timescale long short-term memory neural network for modelling sentences and documents," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 2326–2335, 2015.

[9] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 515–520, 2016.

[10] M. E. Maron, "Automatic indexing: An experimental inquiry," *J. ACM*, vol. 8, no. 3, pp. 404–417, 1961.

[11] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[12] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Machine Learning: ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, 1998, Proceedings*, pp. 137–142, 1998.

[13] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[14] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," *CoRR*, vol. abs/2004.03705, 2020.

[15] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[16] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 785–794, 2016.

[17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 3146–3154, 2017.

[18] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1746–1751, 2014.

[19] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[20] "A implementation of SVM." https://github.com/Gunjitbedi/Text-Classification, 1998.

[21] "A implementation of KNN." https://github.com/raimonbosch/knn.classifier, 1967.

[22] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* Wadsworth, 1984.

[23] "A implementation of CART." https://github.com/sayantann11/all-classification-templates-for-ML, 1984.

[24] J. R. Quinlan, *C4.5: Programs for Machine Learning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[25] "A implementation of C4.5." https://github.com/Cater5009/Text-Classify, 1993.

[26] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory, Second European Conference, EuroCOLT '95, Barcelona, Spain, March 13-15, 1995, Proceedings*, pp. 23–37, 1995.

[27] "A implementation of AdaBoost ." https://github.com/JiangXingRu/Texture-Classification, 1995.

[28] "A implementation of RandomForest." https://github.com/hexiaolang/RandomForest-In-text-classification, 2001.

[29] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 151–161, 2011.

[30] "A MATLAB implementation of RAE." https://github.com/vin00/Semi-Supervised-Recursive-Autoencoders-for-Predicting-Sentiment-Distributions, 2011.

[31] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pp. 1201–1211, 2012.

[32] "A Tensorflow implementation of MV_RNN." https://github.com/github-pengge/MV_RNN, 2012.

[33] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and 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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1631–1642, 2013.

[34] "A MATLAB implementation of RNTN." https://github.com/pondruska/DeepSentiment, 2013.

[35] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1188–1196, 2014.

[36] "A PyTorch implementation of Paragraph Vectors (doc2vec)." https://github.com/inejc/paragraph-vectors, 2014.

[37] "A Tensorflow implementation of DCNN." https://github.com/kinimod23/ATS_Project, 2014.

[38] "A Keras implementation of TextCNN." https://github.com/alexander-rakhlin/CNN-for-Sentence-Classification-in-Keras, 2014.

[39] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pp. 2267–2273, 2015.

[40] "A Tensorflow implementation of TextRCNN." https://github.com/roomylee/rcnn-text-classification, 2015.

[41] M. Iyyer, V. Manjunatha, J. L. Boyd-Graber, and H. D. III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 1681–1691, 2015.

[42] "A implementation of DAN." https://github.com/miyyer/dan, 2015.

[43] "A PyTorch implementation of Tree-LSTM." https://github.com/stanfordnlp/treelstm, 2015.

[44] "A Tensorflow implementation of CharCNN." https://github.com/mhjabreel/CharCNN, 2015.

[45] "A implementation of XGBoost." https://xgboost.readthedocs.io/en/latest/, 2016.

[46] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification," in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 1480–1489, 2016.

[47] "A Keras implementation of TextCNN." https://github.com/richliao/textClassifier, 2014.

[48] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pp. 2873–2879, 2016.

[49] "A PyTorch implementation of Multi-Task." https://github.com/baixl/text_classification, 2016.

[50] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 551–561, 2016.

[51] "A Tensorflow implementation of LSTMN." https://github.com/JRC1995/Abstractive-Summarization, 2016.

[52] "A implementation of LightGBM." https://github.com/creatist/text_classify, 201.

[53] E. Grave, T. Mikolov, A. Joulin, and P. Bojanowski, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pp. 427–431, 2017.

[54] "A implementation of FastText." https://github.com/SeanLee97/short-text-classification, 2016.

[55] T. Miyato, A. M. Dai, and I. J. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[56] "A Tensorflow implementation of Virtual adversarial training." https://github.com/tensorflow/models/tree/master/adversarial_text, 2017.

[57] A. B. Dieng, C. Wang, J. Gao, and J. W. Paisley, "Topicrnn: A recurrent neural network with long-range semantic dependency," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[58] "A PyTorch implementation of TopicRNN." https://github.com/dangitstam/topic-rnn, 2017.

[59] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 562–570, 2017.

[60] "A PyTorch implementation of DPCNN." https://github.com/Cheneng/DPCNN, 2017.

[61] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 4068–4074, 2017.

[62] "A PyTorch implementation of ABSA-PyTorch." https://github.com/songyouwei/ABSA-PyTorch, 2019.

[63] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 1615–1625, 2017.

[64] "A Keras implementation of DeepMoji." https://github.com/bfelbo/DeepMoji, 2018.

[65] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 452–461, 2017.

[66] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 2227–2237, 2018.

[67] "A PyTorch implementation of ELMo." https://github.com/flairNLP/flair, 2018.

[68] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pp. 1063–1072, 2018.

[69] "A Tensorflow implementation of DeepGraphCNNforTexts." https://github.com/HKUST-KnowComp/DeepGraphCNNforTexts, 2018.

[70] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 328–339, 2018.

[71] "A implementation of ULMFiT." http://nlp.fast.ai/category/classification.html, 2018.

[72] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 2321–2331, 2018.

[73] "A Tensorflow implementation of LEAM." https://github.com/guoyinwang/LEAM, 2018.

[74] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: sequence generation model for multi-label classification," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pp. 3915–3926, 2018.

[75] "A PyTorch implementation of SGM." https://github.com/lancopku/SGM, 2018.

[76] Z. Li, X. Ding, and T. Liu, "Constructing narrative event evolutionary graph for script event prediction," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 4201–4207, 2018.

[77] "A PyTorch implementation of SGNN." https://github.com/eecrazy/ConstructingNEEG_IJCAI_2018, 2019.

[78] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, and S. Zhang, "Investigating capsule networks with dynamic routing for text classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 3110–3119, 2018.

[79] "A Tensorflow implementation of TextCapsule." https://github.com/andyweizhao/capsule_text_classification, 2018.

[80] F. Fan, Y. Feng, and D. Zhao, "Multi-grained attention network for aspect-level sentiment classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 3433–3442, 2018.

[81] "A Tensorflow implementation of TextGCN." https://github.com/yao8839836/text_gcn, 2019.

[82] "A Tensorflow implementation of BERT." https://github.com/google-research/bert, 2019.

[83] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4487–4496, 2019.

[84] "A PyTorch implementation of MT-DNN." https://github.com/namisan/mt-dnn, 2019.

[85] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 5754–5764, 2019.

[86] "A Tensorflow implementation of XLNet." https://github.com/zihangdai/xlnet, 2019.

[87] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[88] "A PyTorch implementation of RoBERTa." https://github.com/pytorch/fairseq, 2019.

[89] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[90] "A Tensorflow implementation of ALBERT." https://github.com/google-research/ALBERT, 2020.

[91] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, 2020.

[92] "A Tensorflow implementation of SpanBERT." https://github.com/facebookresearch/SpanBERT, 2020.

[93] "Term frequency by inverse document frequency," in *Encyclopedia of Database Systems*, p. 3035, 2009.

[94] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[95] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543, 2014.

[96] M. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pp. 999–1008, 2010.

[97] A. van den Bosch, "Hidden markov models," in *Encyclopedia of Machine Learning and Data Mining*, pp. 609–611, 2017.

[98] K. Schneider, "A new feature selection score for multinomial naive bayes text classification based on kl-divergence," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21-26, 2004 - Poster and Demonstration*, 2004.

[99] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[100] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pp. 540–545, 2007.

[101] A., P., Dempster, N., M., Laird, D., B., and Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, 1977.

[102] P. Frasconi, G. Soda, and A. Vullo, "Hidden markov models for text categorization in multi-page documents," *J. Intell. Inf. Syst.*, vol. 18, no. 2-3, pp. 195–217, 2002.

[103] K. Yi and J. Beheshti, "A hidden markov model-based text classification of medical documents," *J. Inf. Sci.*, vol. 35, no. 1, pp. 67–81, 2009.

[104] M. O'Donnell, "Cataloging and classification: An introduction," *Technical Services Quarterly*, vol. 26, no. 1, pp. 86–87, 2009.

[105] P. Soucy and G. W. Mineau, "A simple KNN algorithm for text categorization," in *Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA*, pp. 647–648, 2001.

[106] S. Tan, "Neighbor-weighted k-nearest neighbor for unbalanced text corpus," *Expert Syst. Appl.*, vol. 28, no. 4, pp. 667–671, 2005.

[107] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[108] T. Joachims, "A statistical learning model of text classification for support vector machines," in *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pp. 128–136, 2001.

[109] T. JOACHIMS, "Transductive inference for text classification using support vector macines," in *International Conference on Machine Learning*, 1999.

[110] T. M. Mitchell, *Machine learning.* McGraw Hill series in computer science, McGraw-Hill, 1997.

[111] R. J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[112] D. E. Johnson, F. J. Oles, T. Zhang, and T. Götz, "A decision-tree-based symbolic rule induction system for text categorization," *IBM Syst. J.*, vol. 41, no. 3, pp. 428–437, 2002.

[113] P. Vateekul and M. Kubat, "Fast induction of multiple decision trees in text categorization from large scale, imbalanced, and multi-label data," in *ICDM Workshops 2009, IEEE International Conference on Data Mining Workshops, Miami, Florida, USA, 6 December 2009*, pp. 320–325, 2009.

[114] A. Bouaziz, C. Dartigues-Pallez, C. da Costa Pereira, F. Precioso, and P. Lloret, "Short text classification using semantic random forest," in *Data Warehousing and Knowledge Discovery - 16th International Conference, DaWaK 2014, Munich, Germany, September 2-4, 2014. Proceedings*, pp. 288–299, 2014.

[115] M. Z. Islam, J. Liu, J. Li, L. Liu, and W. Kang, "A semantics aware random forest for text classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China,*

November 3-7, 2019, pp. 1061–1070, 2019.

[116] O. Irsoy and C. Cardie, "Deep recursive neural networks for compositionality in language," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2096–2104, 2014.

[117] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using LSTM for region embeddings," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 526–534, 2016.

[118] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling," in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pp. 3485–3495, 2016.

[119] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pp. 1165–1174, 2018.

[120] J. Liu, W. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pp. 115–124, 2017.

[121] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 2915–2921, 2017.

[122] B. Adams and G. McKenzie, "Crowdsourcing the character of a place: Character-level convolutional networks for multilingual geographic text classification," *Trans. GIS*, vol. 22, no. 2, pp. 394–408, 2018.

[123] K. Shimura, J. Li, and F. Fukumoto, "HFT-CNN: learning hierarchical category structure for multi-label short text categorization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 811–816, 2018.

[124] Y. Bao, M. Wu, S. Chang, and R. Barzilay, "Few-shot text classification with distributional signatures," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[125] X. Zhou, X. Wan, and J. Xiao, "Attention-based LSTM network for cross-lingual sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 247–256, 2016.

[126] "NLP&CC Corpus." http://tcci.ccf.org.cn/conference/2013/index.html, 2013.

[127] C. N. dos Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," *CoRR*, vol. abs/1602.03609, 2016.

[128] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional LSTM model and inner-attention," *CoRR*, vol. abs/1605.09090, 2016.

[129] Y. Hao, Y. Zhang, K. Liu, S. He, Z. Liu, H. Wu, and J. Zhao, "An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 221–231, 2017.

[130] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[131] R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, and S. Zhu, "Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 5812–5822, 2019.

[132] J. Chen, Y. Hu, J. Liu, Y. Xiao, and H. Jiang, "Deep short text classification with knowledge powered attention," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 6252–6259, 2019.

[133] T. Shen, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Bi-directional block self-attention for fast and memory-efficient sequence modeling," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[134] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "Large-scale multi-label text classification on EU legislation," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 6314–6322, 2019.

[135] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?," in *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings*, pp. 194–206, 2019.

[136] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 6861–6871, 2019.

[137] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 3442–3448, 2019.

[138] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, P. Yu, and L. He, "Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[139] A. Pal, M. Selvakumar, and M. Sankarasubbu, "MAGNET: multi-label text classification using attention-based graph neural network," in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020*, pp. 494–505, 2020.

[140] M. k. Alsmadi, K. B. Omar, S. A. Noah, and I. Almarashdah, "Performance comparison of multi-layer perceptron (back propagation, delta rule and perceptron) algorithms in neural networks," in *2009 IEEE International Advance Computing Conference*, pp. 296–299, 2009.

[141] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *CoRR*, vol. abs/1704.03976, 2017.

[142] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning – ICANN 2011* (T. Honkela, W. Duch, M. Girolami, and S. Kaski, eds.), (Berlin, Heidelberg), pp. 44–51, Springer Berlin Heidelberg, 2011.

[143] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[144] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 4144–4150, 2017.

[145] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 919–927, 2015.

[146] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pp. 630–645, 2016.

[147] H. Nguyen and M. Nguyen, "A deep neural architecture for sentence-level sentiment classification in twitter social networking," in *Computational Linguistics - 15th International Conference of the Pacific Association for Computational Linguistics, PACLING 2017, Yangon, Myanmar, August 16-18, 2017, Revised Selected Papers*, pp. 15–27, 2017.

[148] Z. Chen and T. Qian, "Transfer capsule network for aspect level sentiment classification," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 547–556, 2019.

[149] W. Xue, W. Zhou, T. Li, and Q. Wang, "MTNA: A neural multi-task model for aspect category classification and aspect term extraction on restaurant reviews," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017, Volume 2: Short Papers*, pp. 151–156, 2017.

[150] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[151] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 606–615, 2016.

[152] Z. Hu, X. Li, C. Tu, Z. Liu, and M. Sun, "Few-shot charge prediction with discriminative legal attributes," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pp. 487–498, 2018.

[153] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

[154] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, "Improved representation learning for question answer matching," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 464–473, Association for Computational Linguistics, Aug. 2016.

[155] A. Radford, "Improving language understanding by generative pre-training," 2018.

[156] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3651–3657, 2019.

[157] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 2978–2988, 2019.

[158] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *CoRR*, vol. abs/1506.05163, 2015.

[159] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3837–3845, 2016.

[160] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 1506–1515, 2017.

[161] Y. Li, R. Jin, and Y. Luo, "Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (seg-gcrns)," *JAMIA*, vol. 26, no. 3, pp. 262–268, 2019.

[162] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an, "Graph convolutional encoders for syntax-aware neural machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 1957–1967, 2017.

[163] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 334–339, 2020.

[164] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[165] L. Hu, T. Yang, C. Shi, H. Ji, and X. Li, "Heterogeneous graph attention networks for semi-supervised short text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 4820–4829, 2019.

[166] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," in *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pp. 737–744, 1993.

[167] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pp. 2786–2792, 2016.

[168] Jayadeva, H. Pant, M. Sharma, and S. Soman, "Twin neural networks for the classification of large unbalanced datasets," *Neurocomputing*, vol. 343, pp. 34 – 49, 2019. Learning in the Presence of Class Imbalance and Concept Drift.

[169] T. Miyato, S. ichi Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," 2015.

[170] T. Zhang, M. Huang, and L. Zhao, "Learning structured representation for text classification via reinforcement learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 6053–6060, 2018.

[171] C. Shen, C. Sun, J. Wang, Y. Kang, S. Li, X. Liu, L. Si, M. Zhang, and G. Zhou, "Sentiment classification towards question-answering with hierarchical matching network," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 3654–3663, 2018.

[172] X. Ding, K. Liao, T. Liu, Z. Li, and J. Duan, "Event representation learning enhanced with external commonsense knowledge," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 4893–4902, 2019.

[173] Y. Zhang, D. Song, P. Zhang, X. Li, and P. Wang, "A quantum-inspired sentiment representation model for twitter sentiment analysis," *Applied Intelligence*, 2019.

[174] "MR Corpus." http://www.cs.cornell.edu/people/pabo/movie-review-data/, 2002.

[175] "SST Corpus." http://nlp.stanford.edu/sentiment, 2013.

[176] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and 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*, (Seattle, Washington, USA), pp. 1631–1642, Association for Computational Linguistics, Oct. 2013.

[177] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, (Barcelona, Spain), pp. 271–278, July 2004.

[178] "TREC Corpus." https://cogcomp.seas.upenn.edu/Data/QA/QC/, 2002.

[179] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pp. 168–177, 2004.

[180] "MPQA Corpus." http://www.cs.pitt.edu/mpqa/, 2005.

[181] "Twitter Corpus." https://www.cs.york.ac.uk/semeval-2013/task2/, 2013.

[182] "EP Corpus." http://www.experienceproject.com/confessions.php, 2011.

[183] Q. Diao, M. Qiu, C. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 193–202, 2014.

[184] "20NG Corpus." http://ana.cachopo.org/datasets-for-single-label-text-categorization, 2007.

[185] "Fudan Corpus." www.datatang.com/data/44139and43543, 2015.

[186] "AG Corpus." http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html, 2004.

[187] C. Wang, M. Zhang, S. Ma, and L. Ru, "Automatic online news issue construction in web environment," in *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pp. 457–466, 2008.

[188] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[189] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1422–1432, 2015.

[190] "Amazon review Corpus." https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products, 2015.

[191] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, and M. Henderson, "The fourth dialog state tracking challenge," in *Dialogues with Social Robots - Enablements, Analyses, and Evaluation, Seventh International Workshop on Spoken Dialogue Systems, IWSDS 2016, Saariselkä, Finland, January 13-16, 2016*, pp. 435–449, 2016.

[192] J. Ang, Y. Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '05, Philadelphia, Pennsylvania, USA, March 18-23, 2005*, pp. 1061–1064, 2005.

[193] D. Jurafsky and E. Shriberg, "Switchboard swbd-damsl shallow-discourse-function annotation coders manual," 01 1997.

[194] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, 2004.

[195] "RCV1-V2 Corpus." http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm, 2004.

[196] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *J. Assoc. Inf. Sci. Technol.*, vol. 63, no. 1, pp. 163–173, 2012.

[197] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "Semeval-2016 task 4: Sentiment analysis in twitter," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016.

[198] F. Wang, Z. Wang, Z. Li, and J. Wen, "Concept-based short text classification and ranking," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pp. 1069–1078, 2014.

[199] "Reuters Corpus." https://martin-thoma.com/nlp-reuters, 2017.

[200] "Reuters Corpus." https://www.cs.umb.edu/~smimarog/textmining/datasets/, 2007.

[201] "NYTimes Corpus." https://catalog.ldc.upenn.edu/docs/LDC2008T19/new_york_times_annotated_corpus.pdf, 2007.

[202] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2383–2392, 2016.

[203] Y. Yang, W. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 2013–2018, 2015.

[204] "Ohsumed Corpus." http://davis.wpi.edu/xmdv/datasets/ohsumed.html, 2015.

[205] "Amazon670K Corpus." http://manikvarma.org/downloads/XC/XMLRepository.html, 2016.

[206] "EUR-Lex Corpus." http://www.ke.tu-darmstadt.de/resources/eurlex/eurlex.html, 2019.

[207] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pp. 115–124, 2005.

[208] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.

[209] https://data.quora.com/First-Quora-Dataset-Release-QuestionPairs.

[210] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pp. 784–789, 2018.

[211] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "MS MARCO: A human generated machine reading comprehension dataset," in *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, 2016.

[212] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 632–642, 2015.

[213] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

[214] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, "Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment," in *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pp. 1–8, 2014.

[215] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland*, 2004.

[216] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*, pp. 364–371, 2017.

[217] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[218] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.